Installer manual

Logic program editor.

By-me Plus SYSTEM

by-me
smart automation

Contents

Contents

Logic program editor

Introduction - General overview

1. Introduction

The Editor is used to create the resident logic programs in the By-me home automation system gateways 01410 and 01411 and the resident logic programs in the 01468 logic unit.

Since the Editor is ONLY accessible from the cloud, Internet connection is always required for the configuration and activation of the programmes.

For all the details relating to management menus of logic programs, see the Logic programs chapter in the By-me Plus System manual.

CAUTION:
In the By-me Plus system, the 01468 logic unit is only used in very rare cases when the device must operate like a PLC without any interaction with the end user (e.g. temperature control applications that require the reliability offered by a dedicated device).
The home automation system gateway and touch screen cannot therefore control the logic unit programs.

2. General overview

2.1 Layout

The following figure shows the structure of the editor's graphic interface, after opening its window:
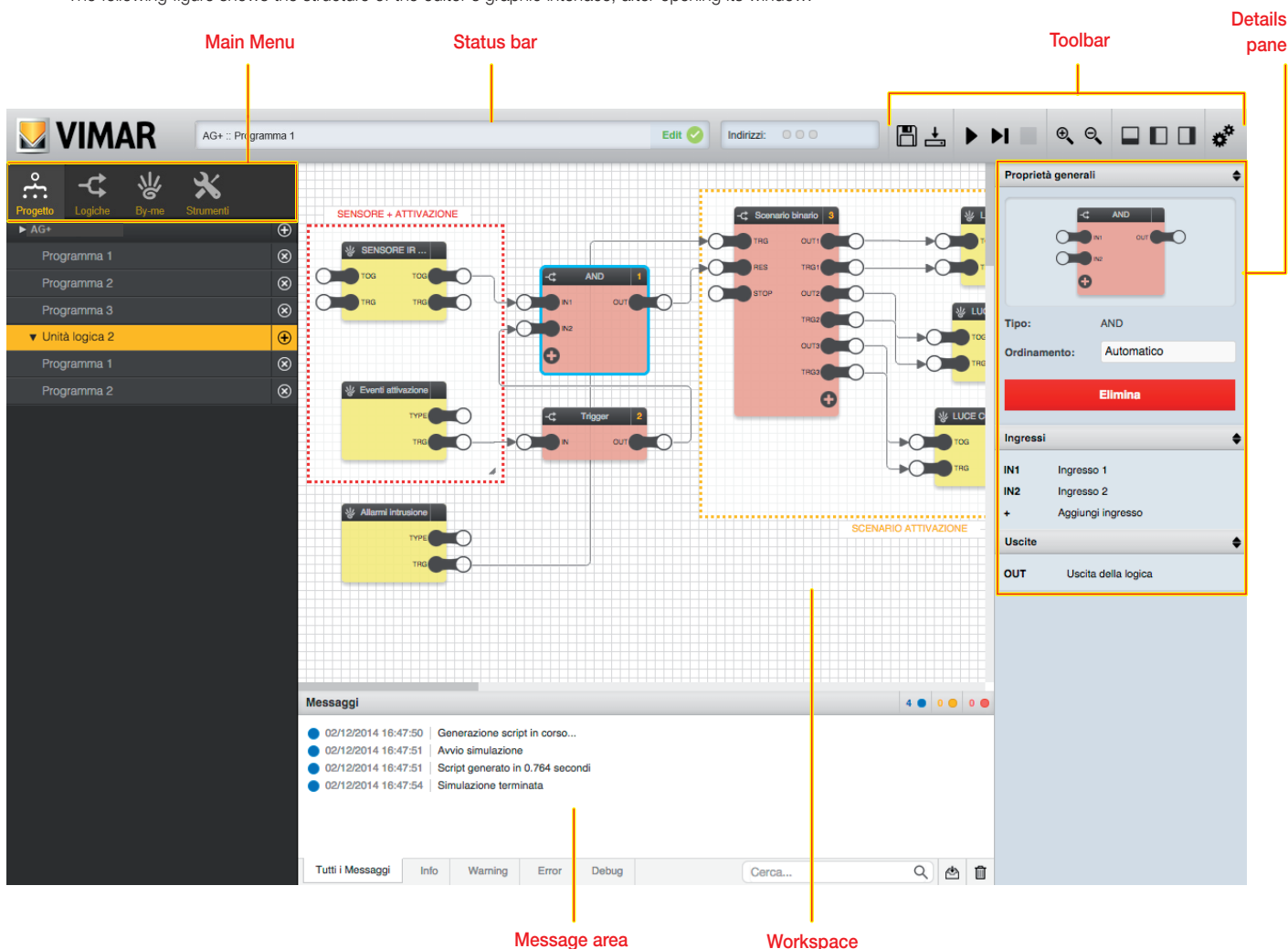
N.B. The icons displayed in the Main Menu and Toolbar vary according to who is using the gateway or the 01468 logic unit editor; the functions, logic blocks and program creation methods are in any case the same.

2.2 Main menu

The menu provides all the tools for creating and managing logic programs. The "tabs" at the start of the menu let you access the main sections of the menu.

Logic program editor

General overview

2.2.1 Progetto

This section of the menu lets you create, modify, and delete logic programs.
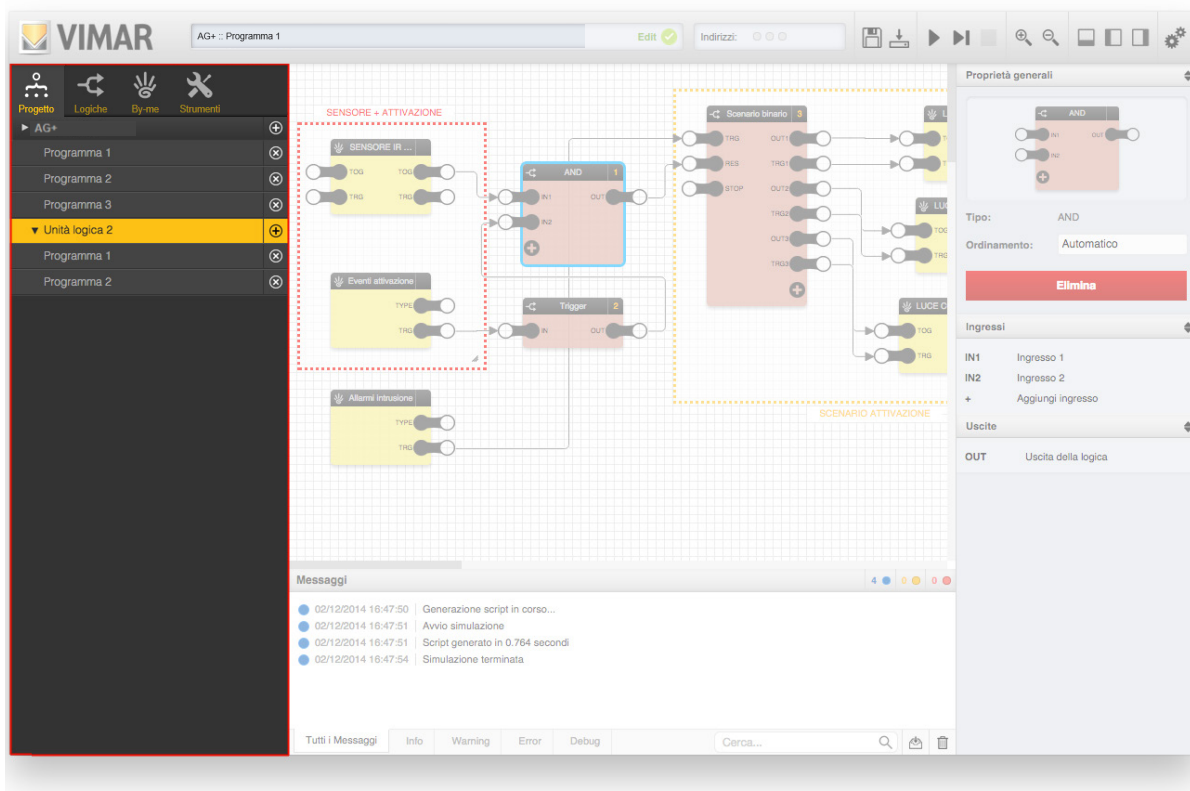
2.2.2 Logiche

This section contains the library of logic blocks that can be inserted in the programs. The items of the logic library can be inserted in the programs by dragging and dropping.

Logic program editor
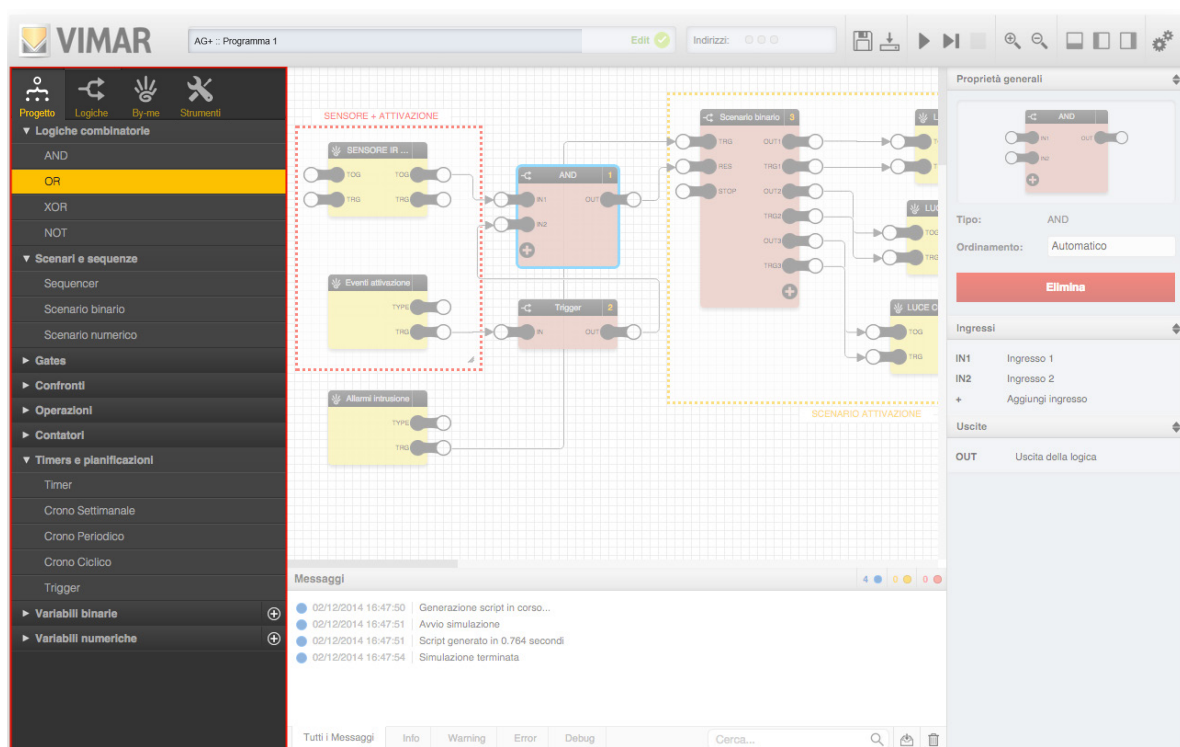
General overview

2.2.3 By-me

This section contains a list of all the groups in the project, divided by type. In this case, too, the By-me blocks can be dragged from this section into the programs to make them interact with the logic elements.
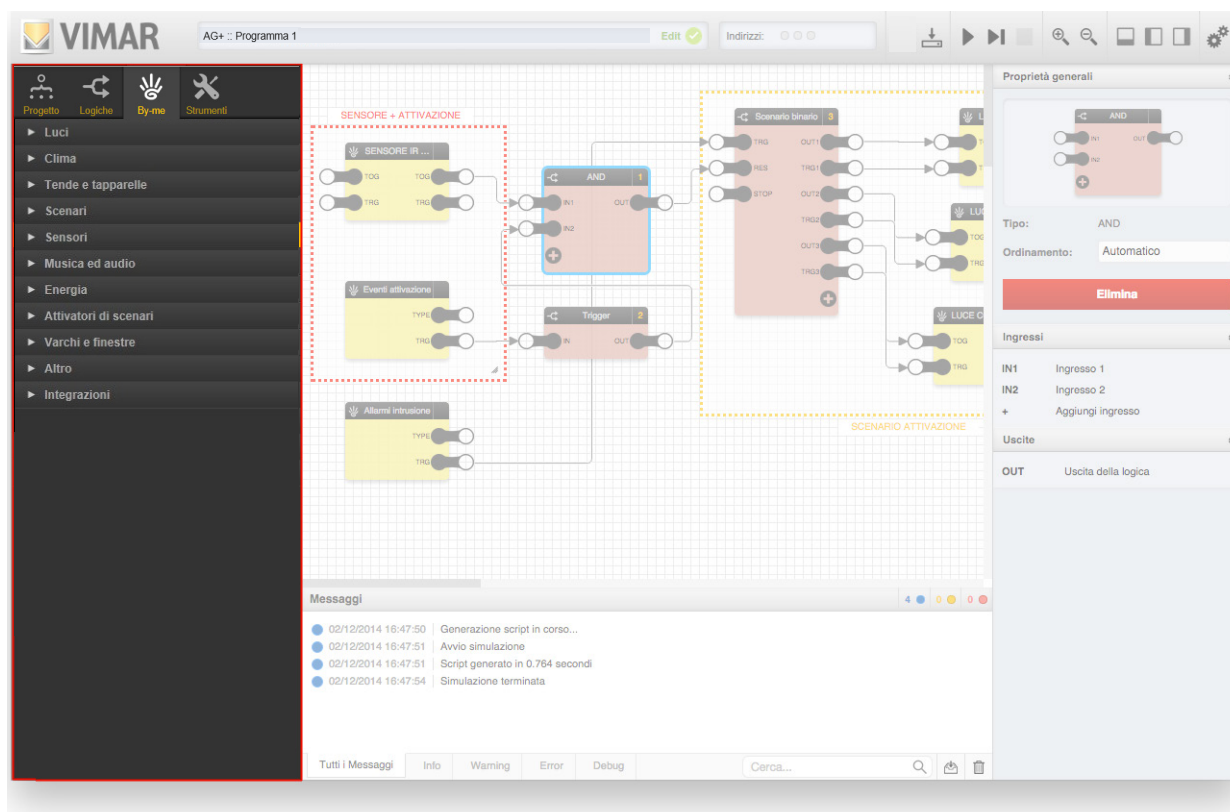
2.2.4 Strumenti

This section lets you insert graphic support elements in logic programs, which can be used to include explanatory comments, notes or boxes with functional groupings, etc.

Logic program editor

General overview

2.3 Status bar

This section of the graphical user interface displays the logic unit and currently selected program, and highlights the current work mode (editing or simulation) as well as any error messages.
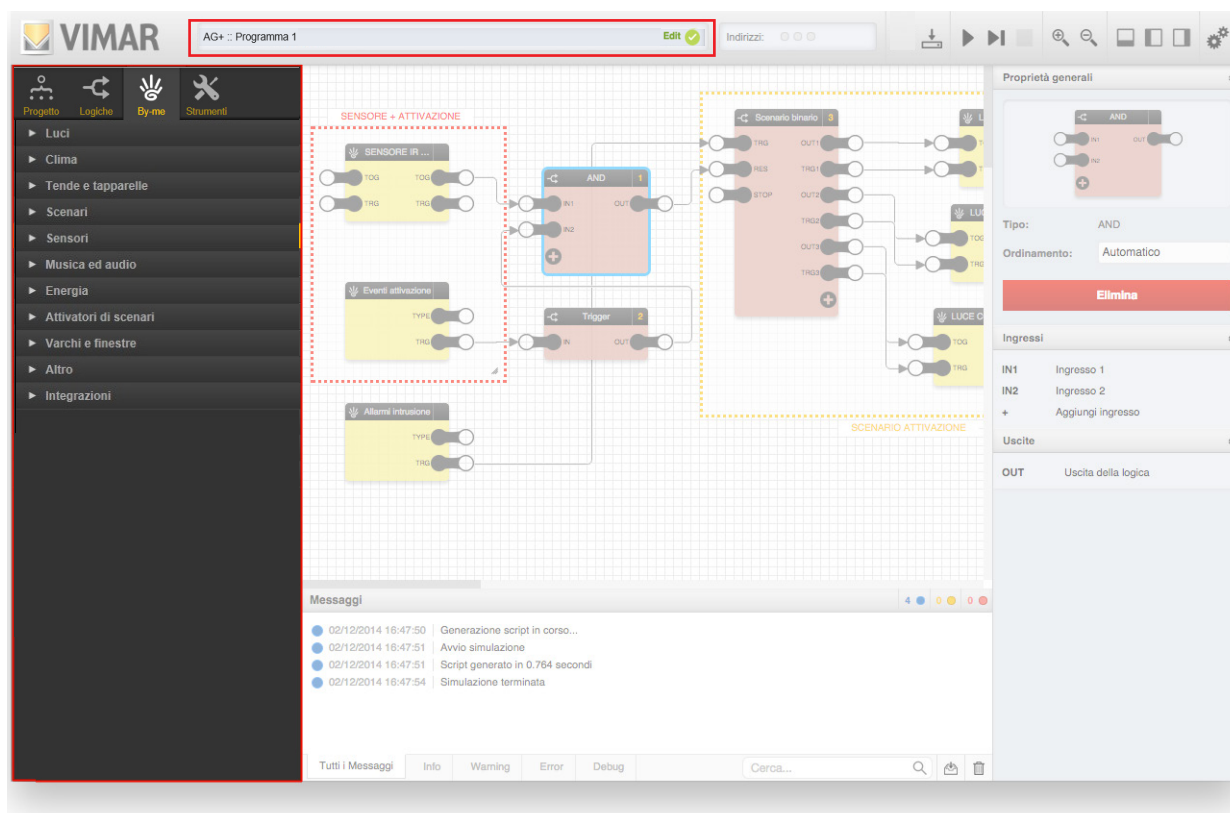
2.4 Used group "indirizzi" bar

This section of the graphic interface displays the group addresses used in the various programs.

Each logic unit can utilize up to a maximum of 254 group addresses.

N.B. This function is only active for programs created with the 01468 logic unit; for those created with the home automation system gateway, the group addresses are not counted.

Logic program editor

General overview

Before compiling files for the logic unit, the calculation is done by using an approximation, the number of group addresses is incremented by one for each connected node. After compilation the group addresses actually used will be calculated. If the logics are created by the home automation system gateway, the group addresses are not calculated.
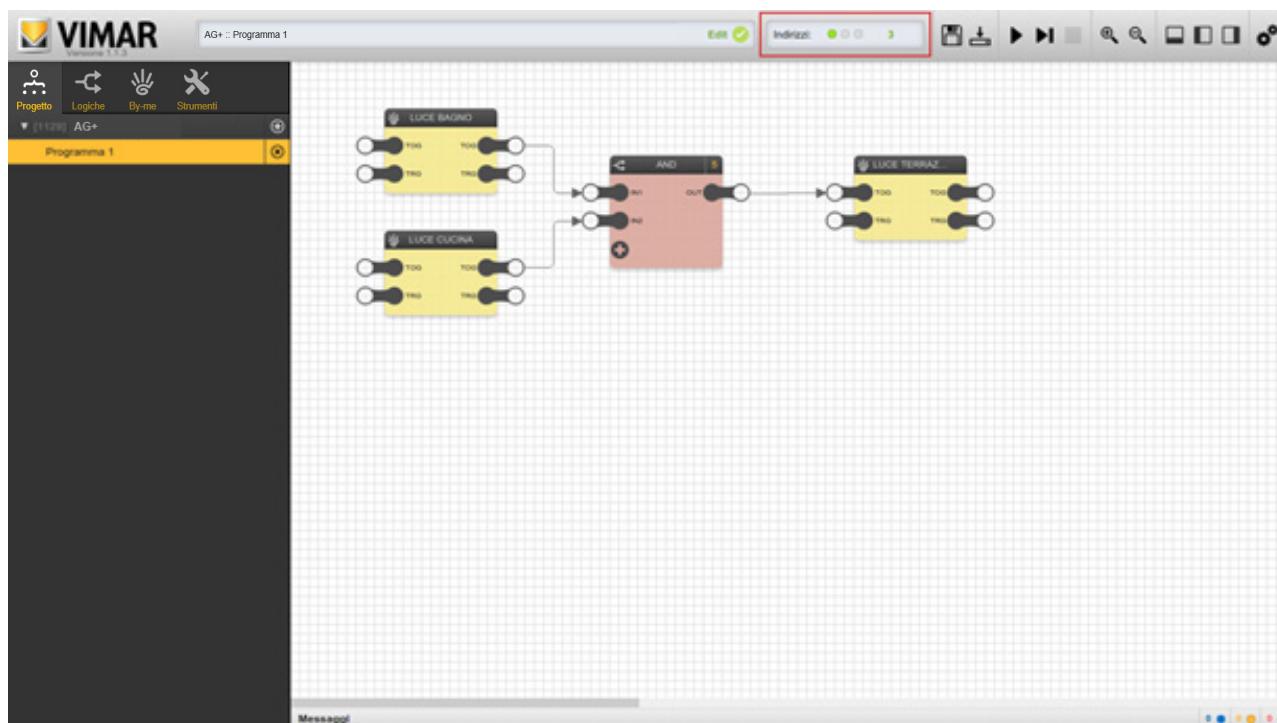
2.5 Toolbar

The toolbar provides the following instruments, which are always available during all running phases of the logic programs (except the simulation phase, when not all of them can be used):

The ADVANCED OPTIONS drop-down menu provides the following items:

SAVE
Saves the logic program configuration
NOTE: Saving is performed automatically also on exiting the graphic environment of the editor.

CONTINUOUS SIMULATION
Starts simulation in real-time mode

STEP-BY-STEP SIMULATION
Starts simulation in step-by-step mode

STOP SIMULATION
Stops the current simulation

ZOOM +
Increases the zoom factor of the workspace

ZOOM –
Decreases the zoom factor of the workspace

SHOW / HIDE MESSAGES
Shows or hides the message area at the bottom

SHOW / HIDE MAIN MENU
Shows or hides the main menu on the left

SHOW / HIDE DETAILS
Shows or hides the side pane containing the details

ADVANCED OPTIONS
Used to access a drop-down menu that contains advanced options, as detailed below

General overview

• Home automation system gateway

Automatic block rearrangement	This regenerates the block sorting in the open program, according to a criterion based on their positions from left-to-right and top-to-bottom. This operation does not overwrite any manually forced sorting on the same blocks.
Editor reload	This forces the editor graphic interface to be redrawn, which is sometimes necessary when the blocks and connection lines are misaligned.
Project reimport	This reimports the files previously loaded on the Cloud by the home automation system gateway.

• Logic units

Used GA recalculation	The group addresses actually used are calculated when the compilation has finished.
Automatic block rearrangement	This regenerates the block sorting in the open program, according to a criterion based on their positions from left-to-right and top-to-bottom. This operation does not overwrite any manually forced sorting on the same blocks.
Editor reload	This forces the editor graphic interface to be redrawn, which is sometimes necessary when the blocks and connection lines are misaligned.
Project reimport	This reimports the files previously loaded on the Cloud.

The Date/Time can be set at any time; specifically, the time can be changed while running a simulation.

It is useful to change the Date/Time during a simulation when the "Orologio Astronomico" (Astronomical clock) and "Temporizzatore Astronomico" (Astronomical timer) are used in order to check the output values of the logic blocks at different set times.
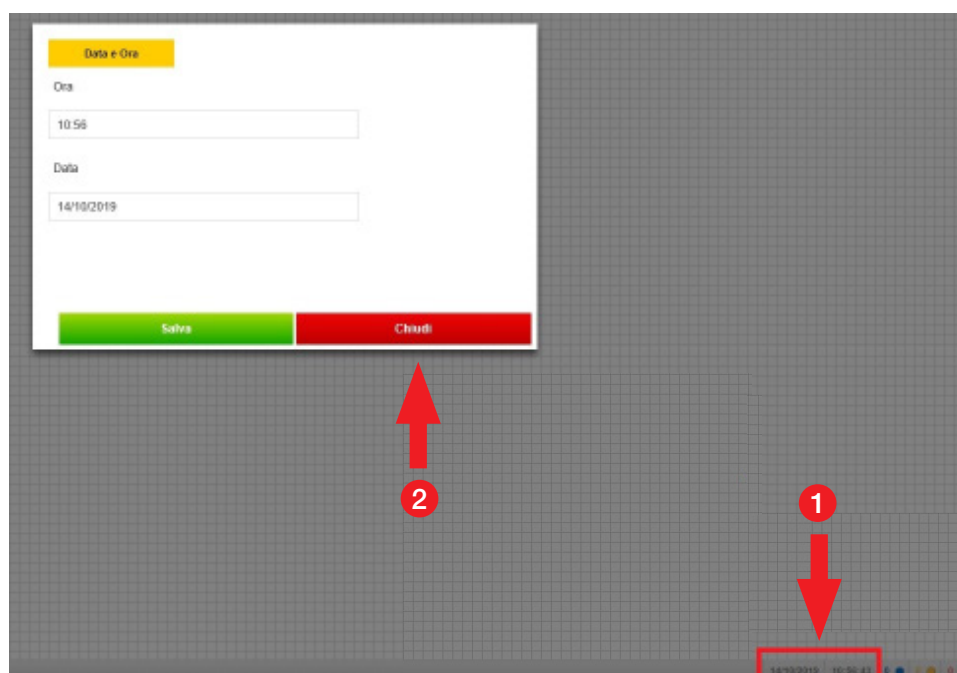
1 Click the highlighted area to display the "Data e Ora" (Date and time) screen.

Logic program editor

General overview

2.6 "Proprietà generali" pane

This section, normally closed (can be opened by using the appropriate button in the toolbar), contains details about the selected objects in the workspace, and lets you change their properties and options. Depending on the type of object selected, the information can be divided into multiple sections, as exemplified in the following figure:
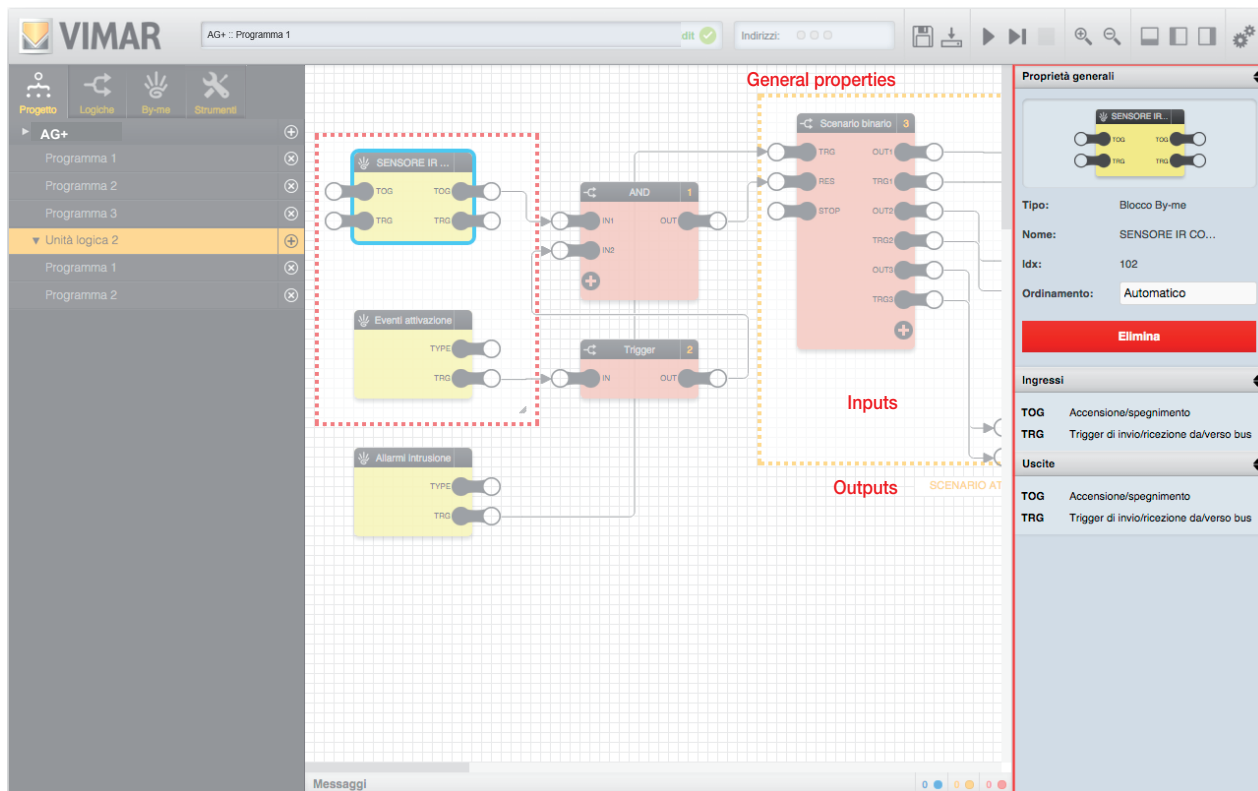
The sections can be closed (by clicking on the title bar) to provide easier consultation of the following ones, especially in the case of objects with many details and options.

2.7 Workspace

The central part of the window is dedicated to the workspace, within which the logics are built. To expand the usable space, it is advisable to close the side panes and the message area, especially when *editing* the logic programs.

Logic program editor

General overview

2.8 "Messaggi" area

The bottom of the window contains the messages generated by the editor during logic program creation and, especially, during simulation (as described below).
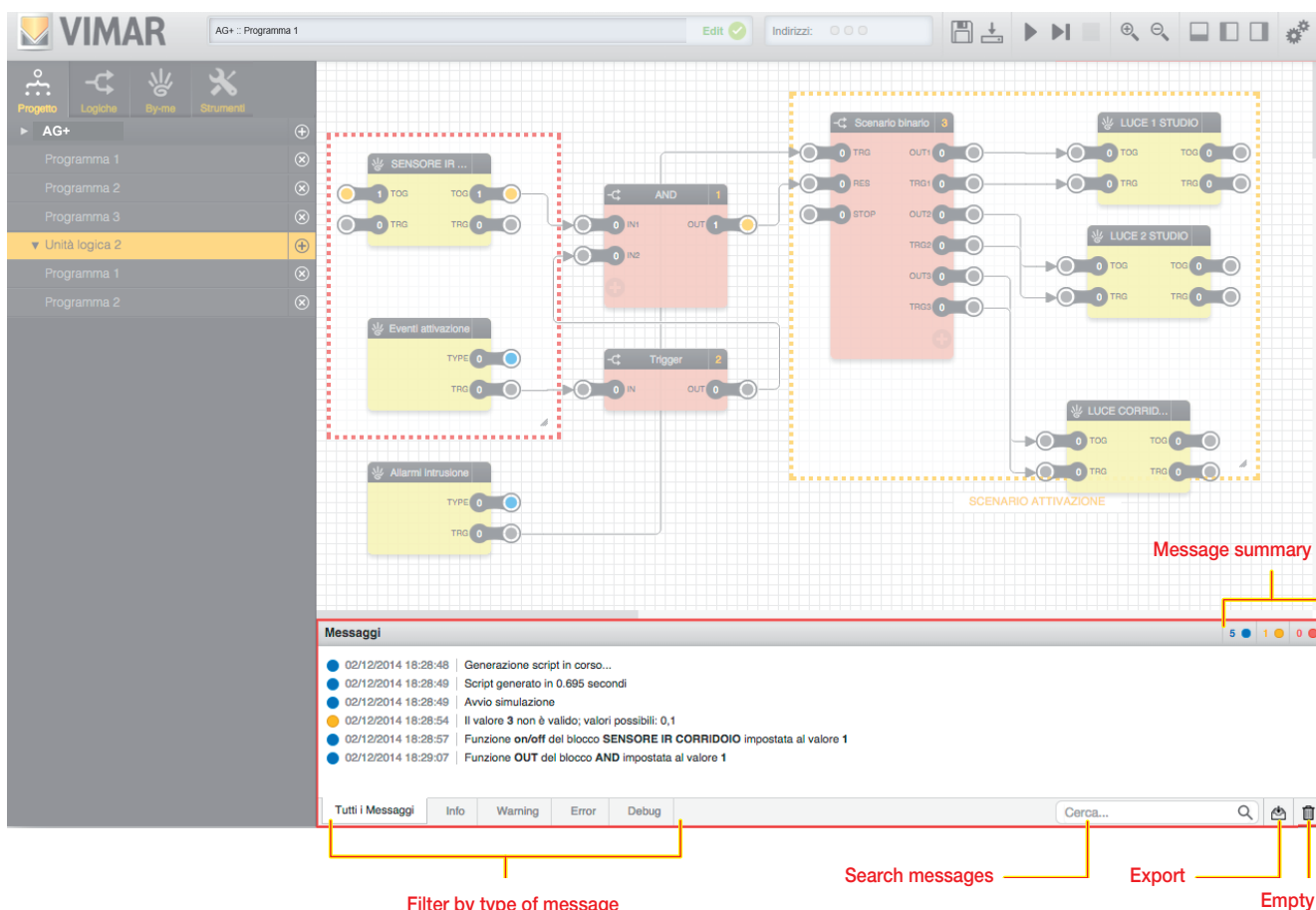
The editor-generated messages can be of different types, according to their severity and type:

- *Error*: warnings of operations or conditions that generate an error, and that typically require editing or checking by the user
- *Warning*: warnings of abnormal conditions, which moreover do not necessarily constitute an error or a situation to be changed
- *Info*: "normal" information messages, which include operations performed by the editor worth signalling to the user
- *Debug*: messages of details of operations carried out by simulation (only available in step-by-step mode, as described below)

The different types are distinguished by a colour, shown beside each message together with the date/time when the message was generated. The title bar of the message area contains a summary, on the right, of the number of messages of the different types, also visible when the message area is closed.

At the bottom of the message area there are the following commands:

- *Filter by message type*: select one of the available items to filter the messages on screen according to the corresponding type
- *Search messages*: used to filter messages based on one or more keywords
- *Export*: used to export the message log (including messages related to previous work sessions) in CSV format, which can also be read with external software (e.g. spreadsheets)
- *Empty*: used to delete video messages (the messages still remain filed in the editor and can be exported via the appropriate button for reading off-line)

Logic program editor

Logic programs

3. Logic programs

3.1 Introduction

Gateways and "unità logica" (logic units) are provided for creating one or more logic networks, each of which is called a "Programma" (Program). These typically receive information from the By-me bus, process it through logic blocks, and send the results over the bus in the form of commands.

A "Programma Logico" (Logic Program) may contain different logics or "functionalities". Theoretically all the functions required may be contained in the same Program; however, dividing the overall logic into several programs offers a number of benefits:

• The Logic Programs can be enabled or disabled from a touch screen or mobile device (if you associate a function to a Program you can then enable or pause that function remotely);

• Maintenance and any modifications after the first draft are made easier (more order).

It should be noted however that attention must be paid to potential interactions and overlaying between different Programs (e.g. use of the same By-me resources in more than one program with possible conflicts such as an actuator controlled by more than one logic program).

Each logic unit can contain up to a maximum of 64 programs.

• Each logic unit can manage up to a maximum of 254 group addresses. You must check the limit before downloading the programs on the logic unit.

   If the limit is exceeded, once the program has been compiled a message error is displayed in the debug area and in the Edit box at the top where the program name appears.

The editor lets you configure the logic programs, connecting By-me blocks and logic functions using *drag&drop* and simple graphic tools, without any particular programming skills. As will be shown later, the editor also enables simulating the behaviour of logic programs, before "downloading" the programming into the logic unit itself.

CAUTION:

In the By-me Plus system, the 01468 logic unit is only used in very rare cases when the device must operate like a PLC without any interaction with the end user (e.g. temperature control applications that require the reliability offered by a dedicated device).

The home automation system gateway and touch screen cannot therefore control the logic unit programs.

3.2 Creating a new program

To create a new program, enter the "PROGETTO" section of the main menu and press the "+" button; a new program called "Programma 1" will be created.

To open the new program, simply click on it: the workspace shows an empty grid, on which you can begin constructing the logic, as described below.

To change the name of the program, open the "Proprietà generali" (details) pane and enter the new name in the text box, as shown in the following figure; the name cannot contain special characters and must have a maximum length of 16 characters.

Logic program editor
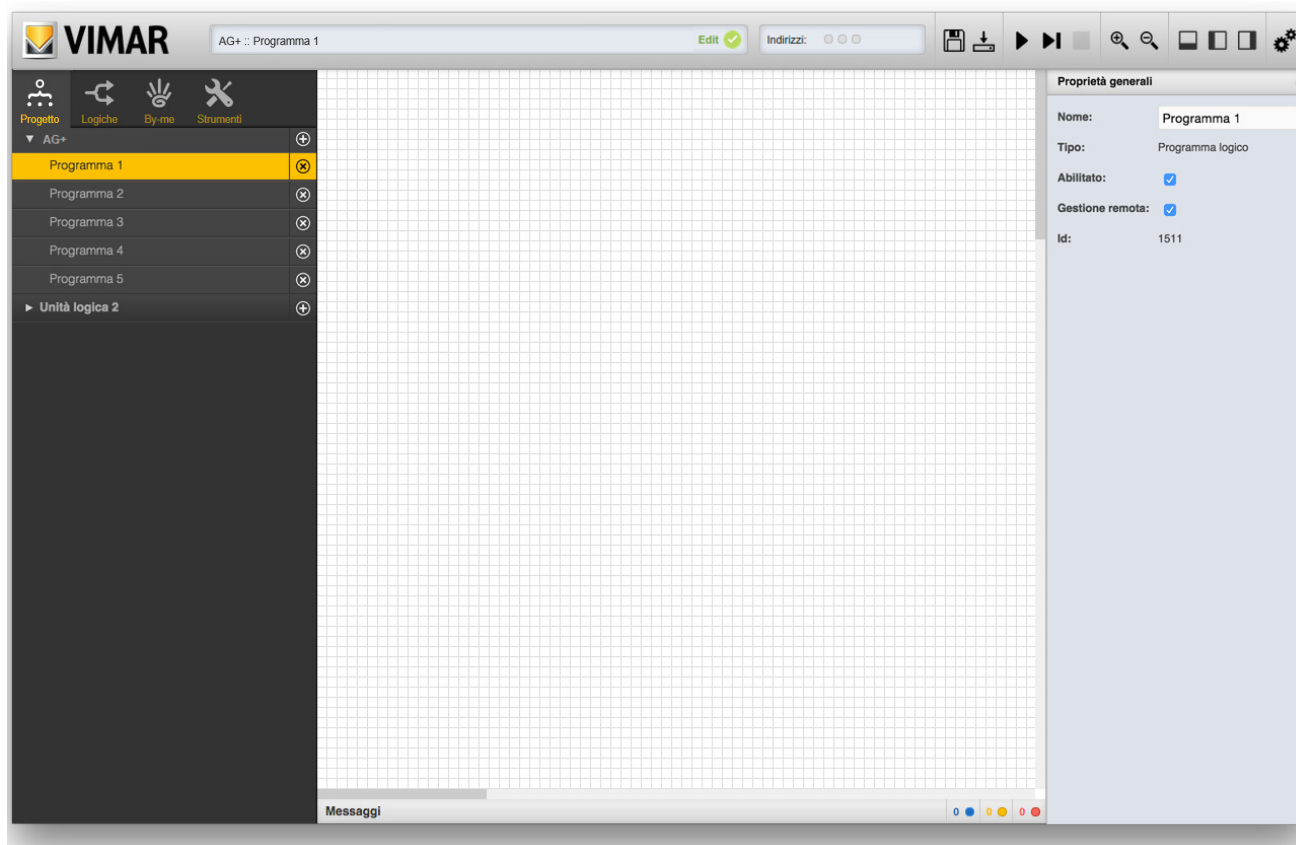
Logic programs

3.3 Removing or disabling a program

To remove an existing program, simply press the corresponding "X" button in the main menu; after confirming deletion, the program is deleted along with all the logic functions it contains. This operation cannot be undone.

If you do not want a program to be included in the logic unit, for example because it is still incomplete, you can disable it by unchecking the corresponding "ABILITATO" item in the "Proprietà generali" pane; disabled programs are shown in the main menu with a semi-transparent effect.

Disabled program effect   Program activation flag
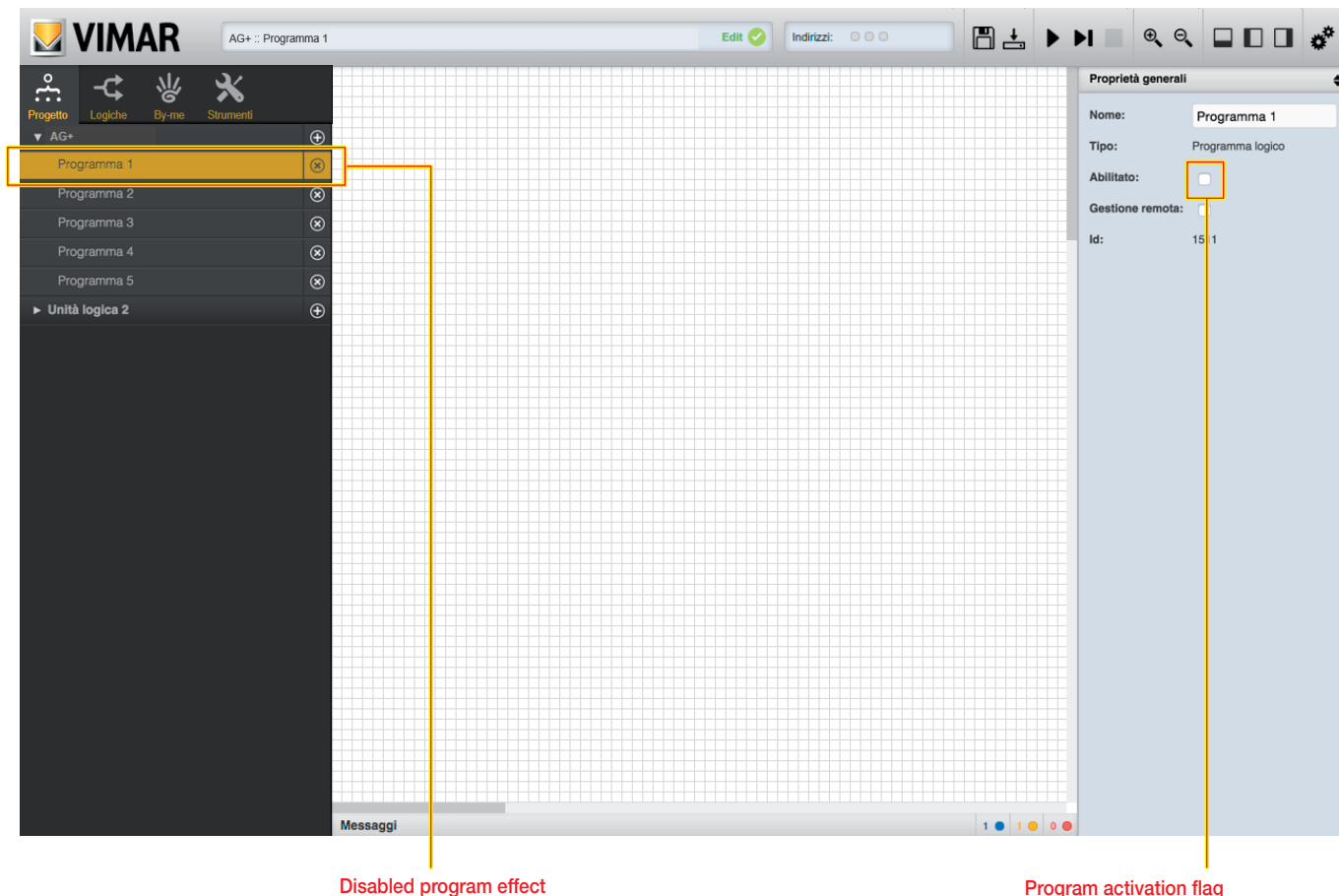
3.4 Remote management

As explained below, the user can manage logic programs via a mobile device or touch screens, but this functionality can also be disabled (for example because the program must not be disabled, or contains timer programming that must not be modified by the user).

3.5 Adding blocks to a program

Programs provide for connecting several blocks to form a logic network. Blocks can be of the By-me or logical type; the former are required to read and/or write information on the home automation bus, the latter enable processing and combining this information.

To add a By-my block to a program, first identify it in the section of the same name in the main menu; here there is a list of all the supported By-me groups.

Logic program editor
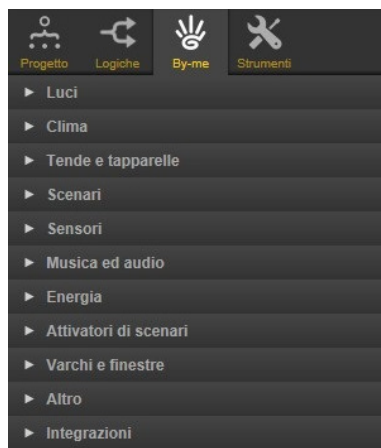
Logic programs

After identifying the By-me block, simply drag it into the workspace by using *drag&drop*:
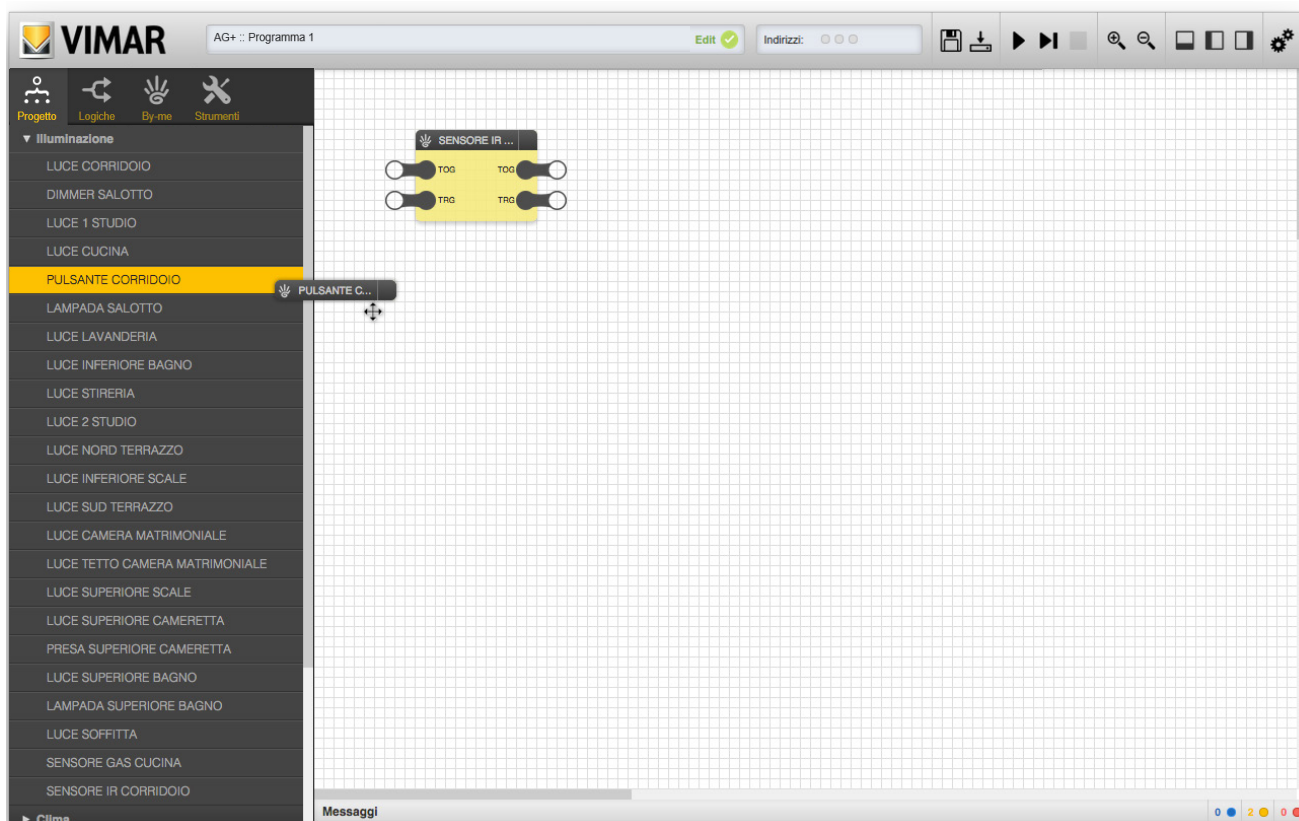
Similarly, to add a logic block, first identify it in the "LOGICHE" library, which is also organized by type (for a complete list of the available logic blocks, see chapter 5), and then drag it into the workspace:

Logic program editor
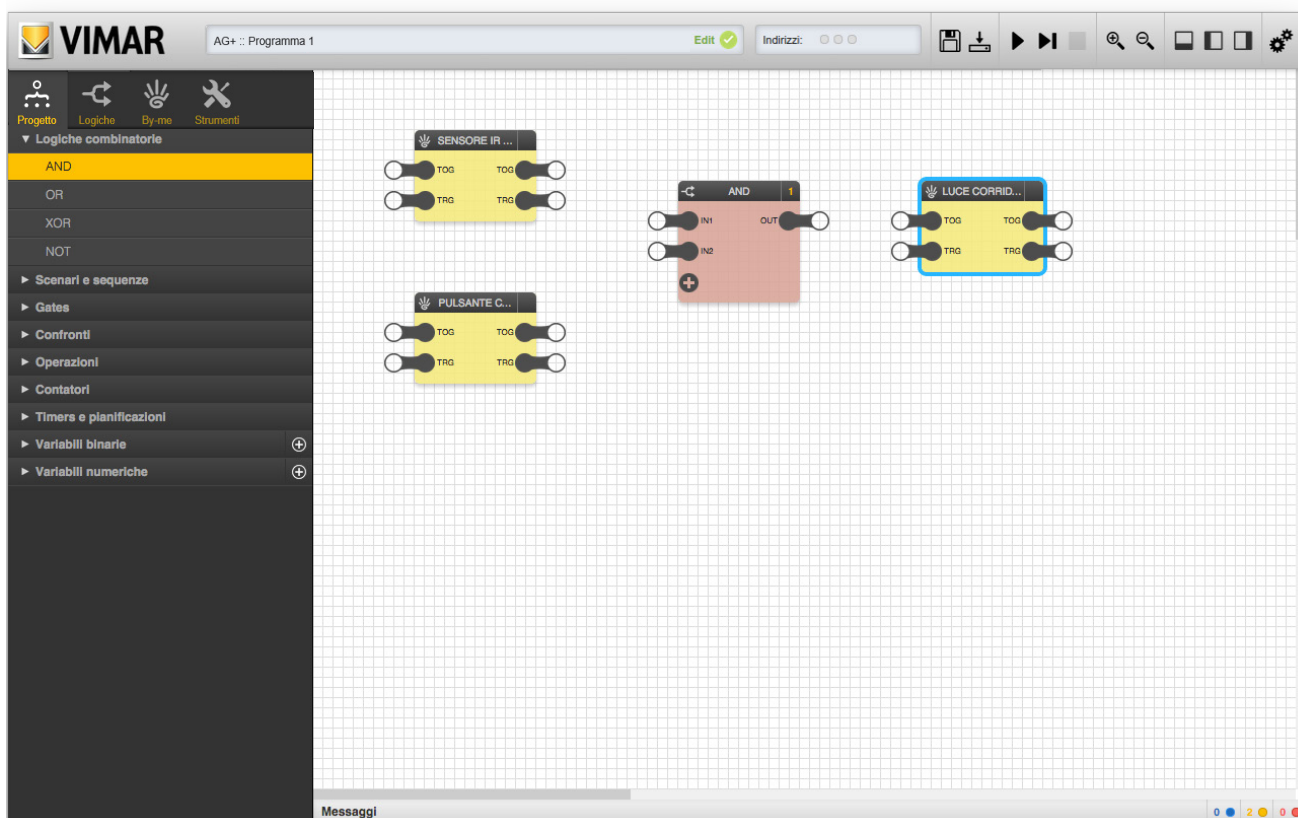
Logic programs

3.6 Selecting one or more blocks

You can select one or more blocks within a program in several ways:

- Clicking on the "title" of the block (single selection)
- Clicking on the "title" of multiple blocks, while holding down the CTRL key ("scattered" multiple selection)
- Clicking on a point in the workspace and, while holding it down, moving the cursor to draw a rectangular selection area ("contiguous" multiple selection)

The selected blocks are displayed with a light blue border:
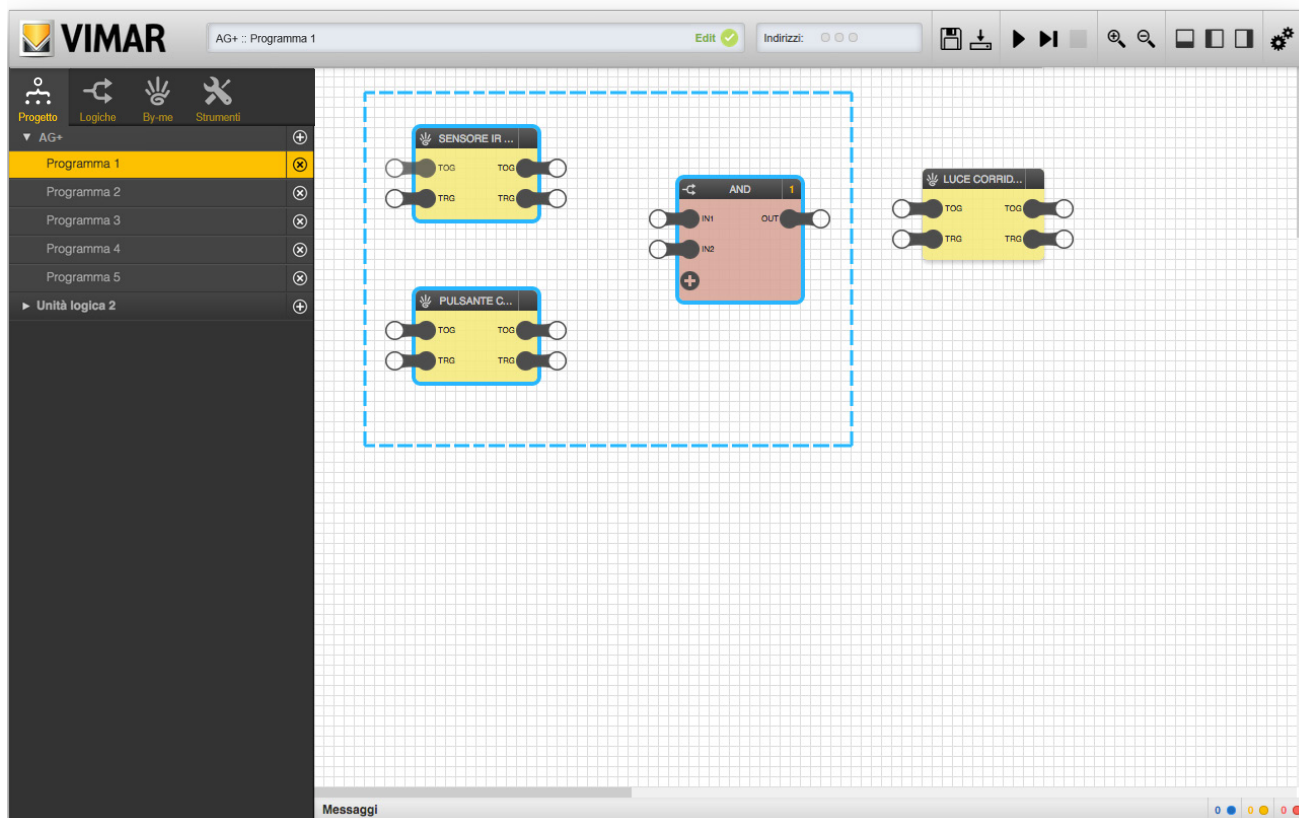
The selected blocks can be moved within the workspace by simply using *drag&drop*. Selecting a single block and opening the "Proprietà generali" pane displays its properties and a list of input and output nodes, and it is possible to manage any options there may be, as detailed below for each type:

Logic program editor

Logic programs
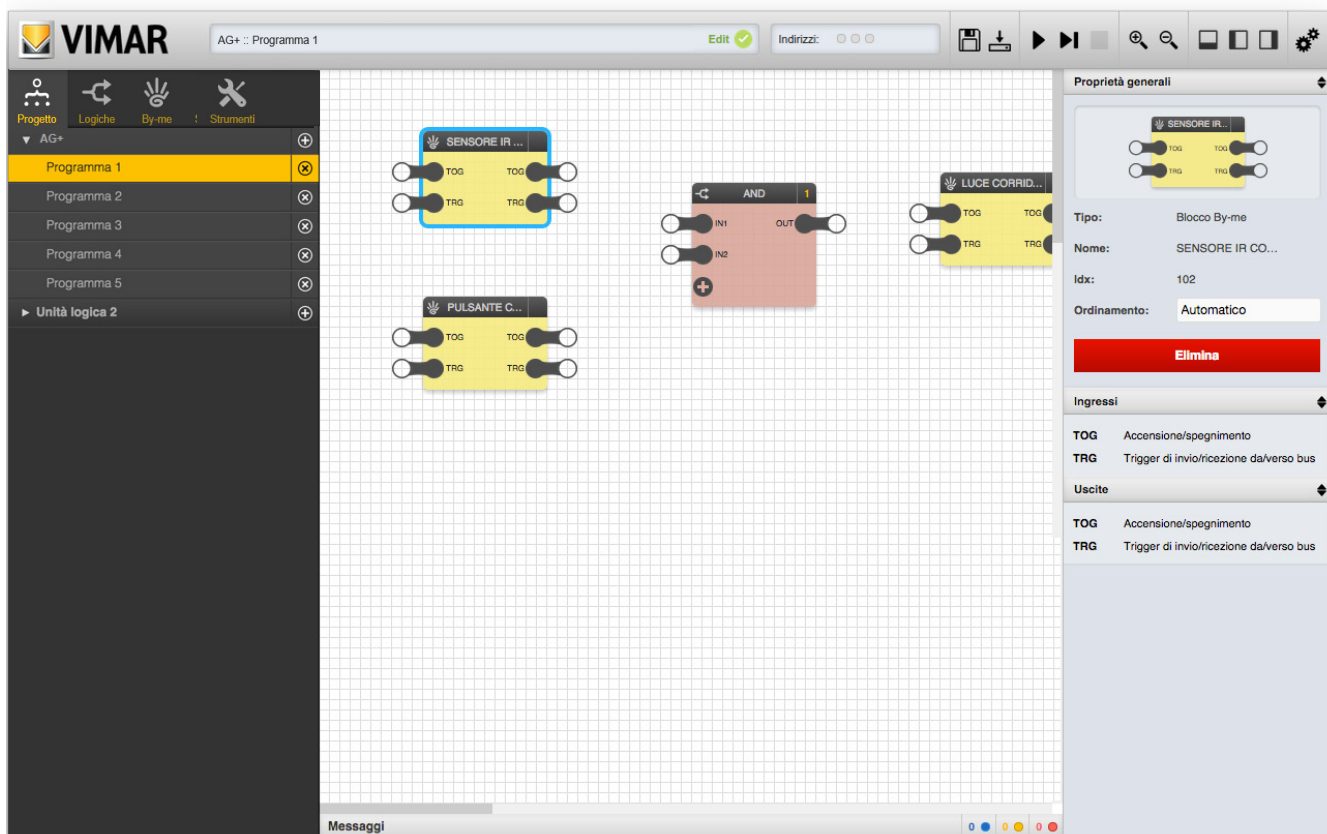
NOTE: When selecting multiple blocks simultaneously you cannot see the details, as they are different for each one of them.
Note that sorting, already explained for the logic blocks, is also available for By-me objects; this advanced configuration element is provided for By-me objects, but it is currently reserved for future use.

3.7 Removing one or more blocks

To remove one or more blocks of a program, proceed in one of the following ways:
- Select a single block, open the "Proprietà generali" pane and press the "ELIMINA" button
- Select one or more blocks and press the "DEL" key on the keyboard

In both cases, after a confirmation message, the selected blocks are removed from the program, as well as any connections with other blocks in the program. This operation cannot be undone.

3.8 Input and output nodes

Each block contains at least one input and/or output "node", as shown in the following figure:
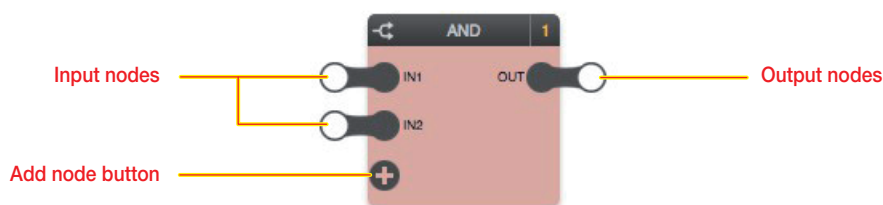
The input nodes are always on the left side of a block, while the output nodes are on the right. Each node is characterized by a concise tag (e.g.: "IN1", "IN2" and "OUT" in the figure above), which appears in the inputs/outputs list in the "Proprietà generali" pane, along with a brief description of each node (as well as in this manual).

Logic program editor

Logic programs

3.8.1 Logic blocks

In the case of the logic blocks, the input nodes represent the "inputs" for the logic function associated to the block, while the output nodes are the "outputs":
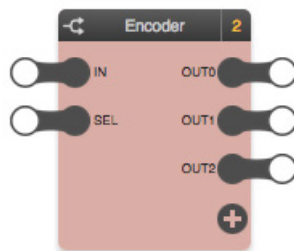
In some cases, as in this example, the block provides for a variable number of nodes (input or output nodes); in this case, the "+" button is used to add nodes to the block, up to the maximum number.

The logic function can only be run properly if the input nodes are connected to other blocks (both logic and By-me blocks) and if the output values are "reported" on the input nodes of the same number of blocks (both logic and By-me blocks).

Not all the input nodes are strictly necessary to exceecute the logic function correctly; if an input node is not connected, its default value is used. This can be changed by selecting the node and opening the corresponding "Proprietà generali" pane, as shown in the following figure:

Selected node   Default value

The detail pane of a node also shows the possible values it can have; this information can be useful especially for blocks that provide for specific value combinations or restrictions.

The logic blocks may also have outputs only, as shown in the following example (scheduling block):

In this case, they can only be used as an input for other logic elements, but they cannot be controlled. In the specific case of scheduling, for example, as detailed later (section 6.9.2), the value depends on the system clock of the logic unit, according to preset programming.

3.8.2 By-me blocks

In the case of By-me blocks, the input nodes (left side) represent the possible commands that the logic unit can send to the corresponding group (bus transmission); conversely, the outputs (right side) are the states that the logic unit can receive from the corresponding group via the bus. In the case for example of a "dimmer" group...

... a number of different data types are available as either input or output:

- TOG ("toggle"): switching the dimmer on / off
- VAL ("value"): dimmer percentage
- LONG.P: long press start/end
- DIR: long press direction

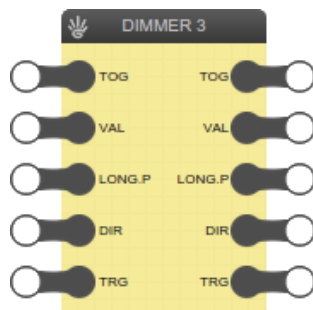If you want to send one of these two values over the bus, you need to connect the output of the corresponding logic element to the input node (left side), so that whenever the logic changes its value it gets sent, via the bus, to the corresponding dimmer actuator. Vice versa, if you want to build a logic element that is based on the state of one of these two pieces of information, it is necessary to connect the corresponding output node (right side) to one or more logic blocks, so that each change in status detected by the bus is "passed" on to the logic element.

Not all By-me blocks have the same number of input and output nodes; some information can only be read (e.g.: temperature measured by a thermostat) or, vice versa, some commands can only be sent to the devices but have no meaning as regards status (e.g.: moving or stopping a roller shutter).

3.8.3 Trigger

By-me blocks provide for, as either input or output, a special "trigger" node (TRG):

- INPUT TRIGGER (left side): used to force transmission of the values of the input nodes (connected to other logic elements) even without a change in value.
- OUTPUT TRIGGER (right side): used to detect data reception from the bus on one of the output nodes, even if it is the same as the previous data (i.e. with no change in value).

In both cases, the trigger is normally 0 and goes to 1 when the trigger is active:

- INPUT TRIGGER: when set to 1, it forces transmission (once) until it is reset and reset back to 1 (or there is a change in value).
- OUTPUT TRIGGER: set to 1 by the logic unit whenever data is received from the bus related to the corresponding group of the By-me block in question (on one of the output nodes of the block) and is then automatically reset with the next cycle.

Under normal circumstances, the logic blocks do not use triggers, but operate on a change of state; where necessary, however, specific logic blocks also have input/output trigger nodes, which operate in the same way as described above for the By-me blocks.

A trigger node (TRG) must be connected to a node with the same operating method (status change pulse); if this is not the case, the specific logic block must be used.

3.9 Connecting blocks

For the program to perform effectively, you must include at least one "connection" between two nodes of two blocks, so that the value of the first one ("source") is passed to the second one ("target"). To connect two nodes, simply click in the centre of the source node, hold and release in the centre of the target node:

Logic program editor

Logic programs
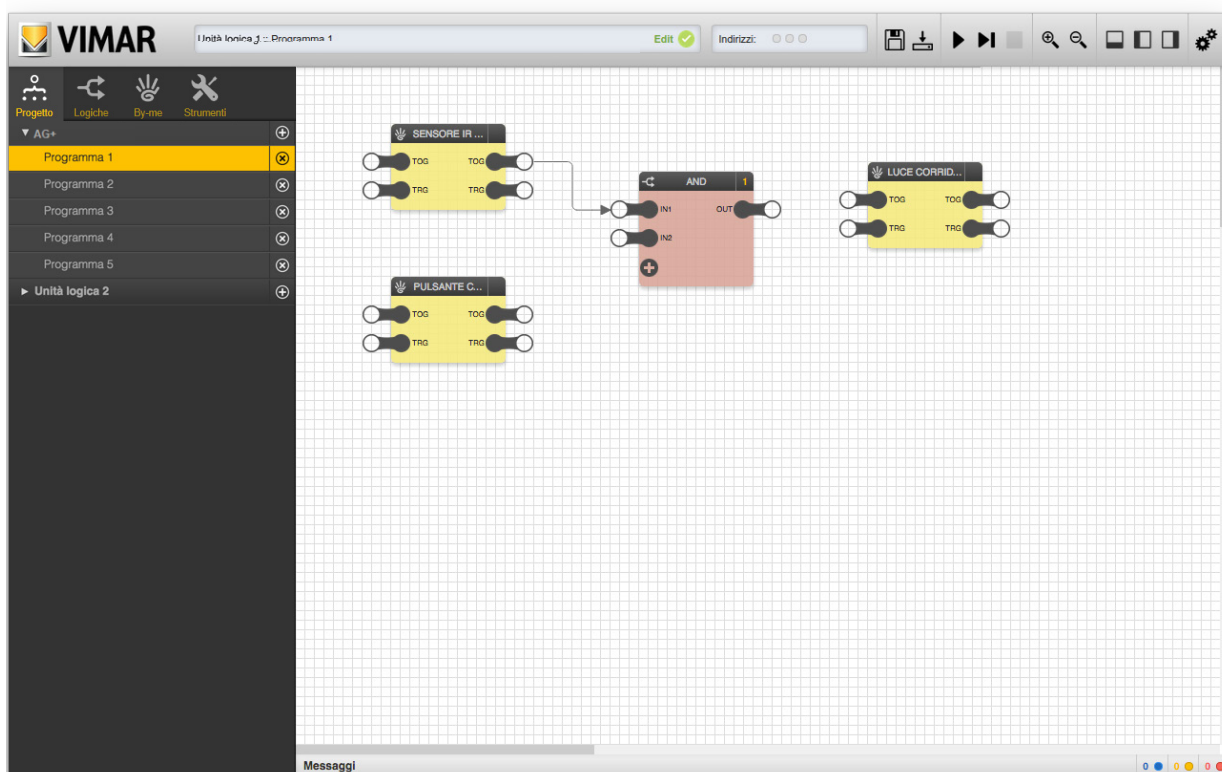
Moving the cursor over a link highlights it in red (and passes "on top" of any other connections or blocks along its route); clicking on it, conversely, selects the link:
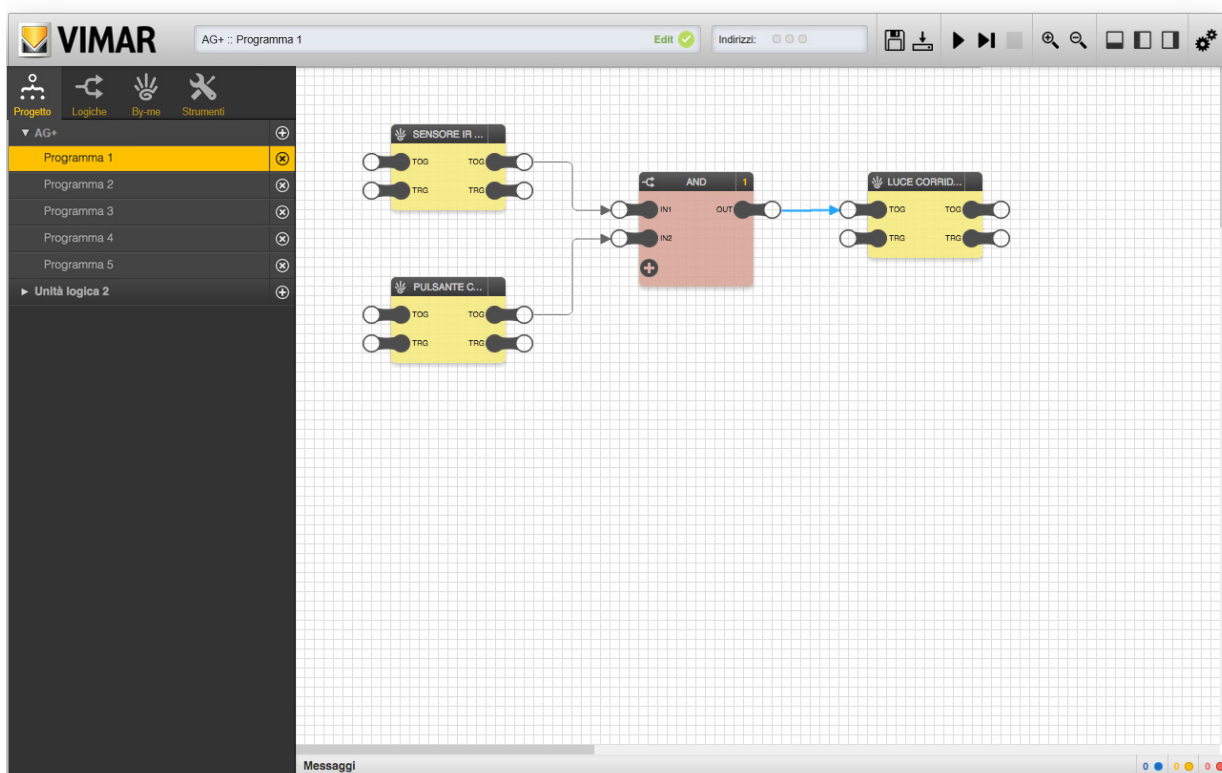
The selected link can be deleted in either of two ways:
- Pressing the "ELIMINA" button in the "Proprietà generali" pane
- Directly pressing the "Del" key on the keyboard

The source of a connection must be an output node (right side of a block) while the target must be an input node (left side); an output node can be the source of several connections (with different targets), while an input node can be the target of only one connection:

Logic program editor

Logic programs
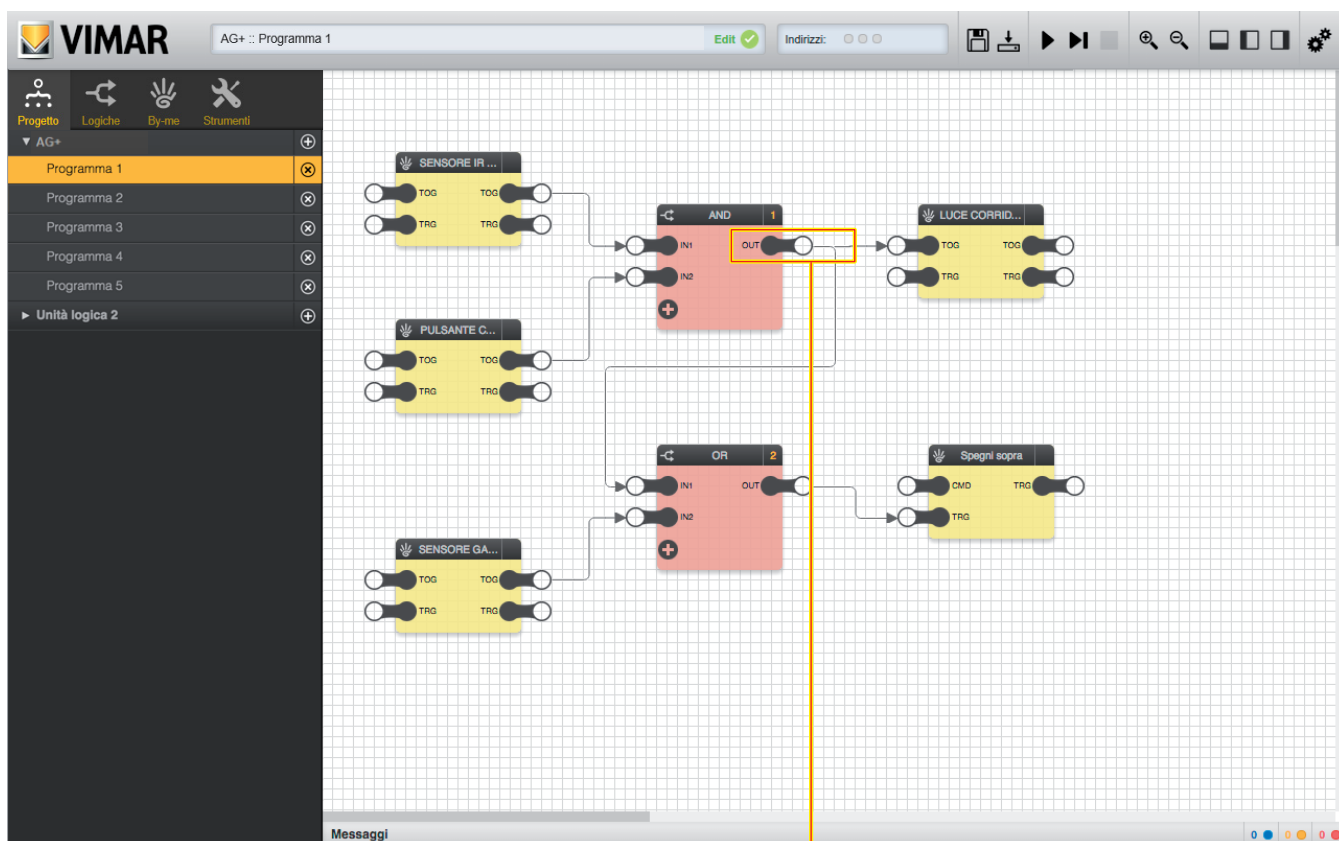
Output node with more than one connection

3.10 Types of nodes

The following table gives the types of nodes.

Type of node	Description
T	TRIGGER: The change in the node value is instantaneous, the node value immediately returns to the value prior to this change.
S	STATUS: The value keeps stable until the next change in status.
M	MIXED: Independent node on status change; it can be of either the STATUS or the TRIGGER type.

When connecting nodes together it is important to pay attention to their type: you cannot directly connect a TRIGGER node to a STATUS node and vice versa. Instead, you can connect STATUS or TRIGGER nodes to MIXED nodes.
Thanks to these types the application allows you to make no connection errors.

3.11 Execution order

While simulating and compiling, as detailed below, the editor, starting from the graphically designed logic networks, generates a "list" that is put into execution in a cyclic manner, from beginning to end, as quickly as possible (depending on the complexity of the project).

3.11.1 Program order

On each run cycle, the following operations are performed (the cycle time depends on the number and complexity of the programs):

- Read inputs from bus
- Run program 1
- Run program 2
- …
- Run program n
- Write commands on bus

The order of the programs is exactly as shown in the main menu; this implies that any interactions between the programs (such as passing values by means of variables, or multiple programs writing the same node of a By-me block) are affected by this order (and any actions taken by the programs "at the end" of the list are received by the previous ones only in the next cycle).

NOTE: If a program is disabled (see 3.3) or paused, it is "skipped" in the cycle; in this case, any interaction with the bus and/or with other programs is suspended.

Logic program editor

Logic programs

3.11.2 Block order

Within each program, the logic blocks also have their own execution order; the logic unit processes the function associated with the logic blocks in the following order. The order of a logic block is shown at the top right, as in the following figure:

Execution order
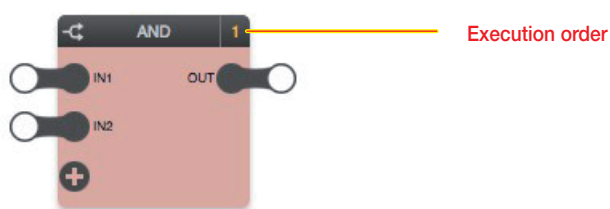
IMPORTANT: Always make sure that the order of the blocks is in line with the order of the logic performance (otherwise the logic will malfunction).

Under normal circumstances, the blocks are assigned an ascending order based on the order in which they are added to the program; however, it is possible to force a different order in the following way:

- Select the block concerned
- Open the "Proprietà generali" pane
- Select "MANUALE" for the "ordinamento" (sorting order)
- Enter an order number, making sure to enter a number that has not yet been used

Blocks with manual sorting are highlighted as follows:

Manual execution order
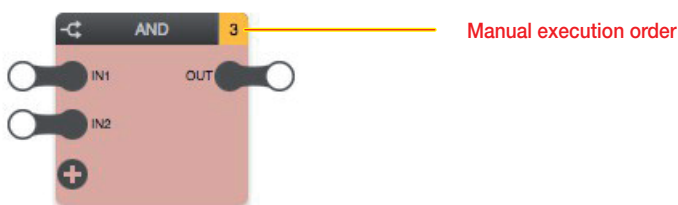
The following figure shows an example of a logic network with a block in manual sorting order, showing how to change the execution order of the blocks:
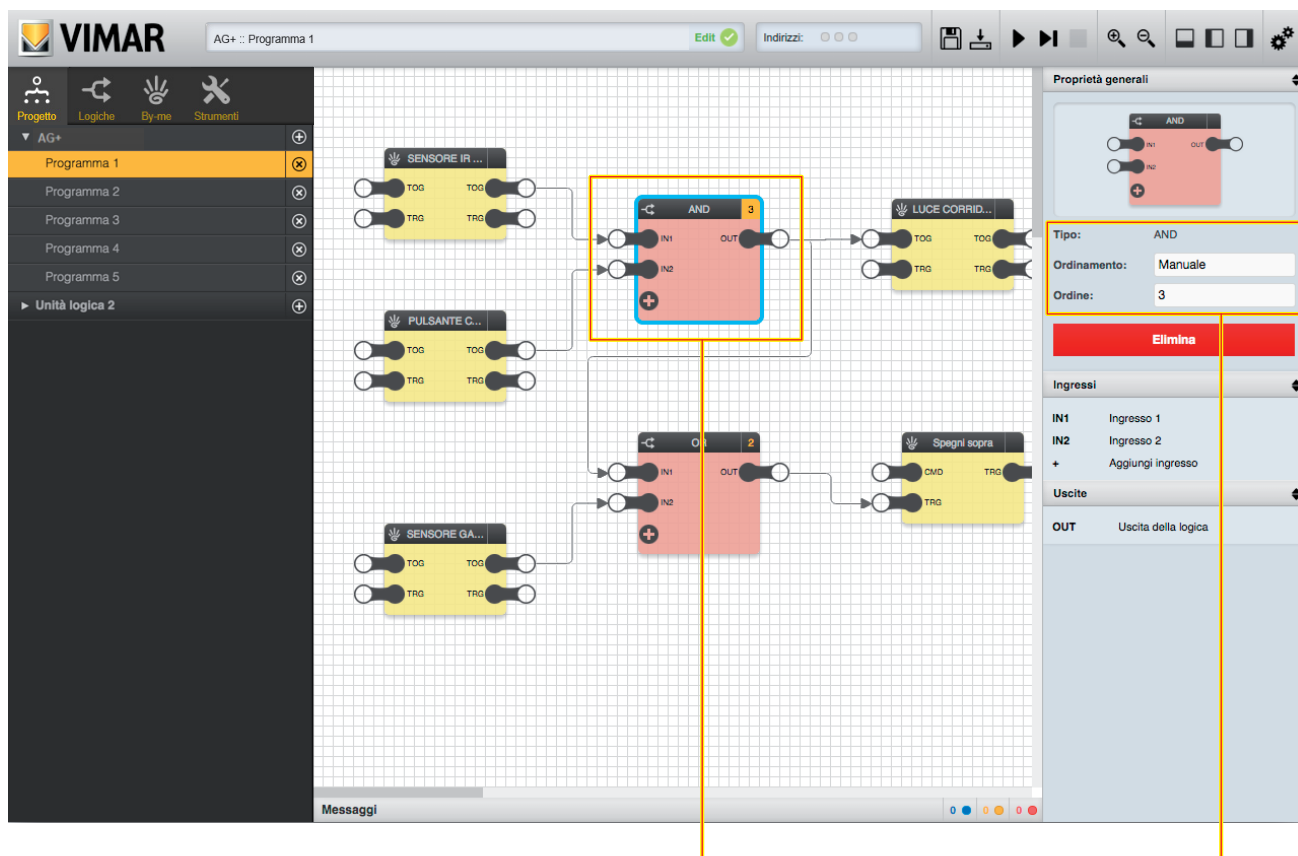
Block with manual execution order

Block execution order settings

Logic program editor

Logic programs

The By-me blocks do not have an execution order, their sorting is not currently important and is reserved for future functions.
They do not represent processing by the logic unit, but only points for reading and writing via the bus; as mentioned previously, the states of the output nodes of all the By-me blocks (of all the active programs) are read at the beginning of each cycle and the commands to the input nodes of all the By-me blocks (of all the active programs) are sent over the bus at the end of the cycle, regardless therefore of the position of the blocks in the programs and of the order of the programs.

Generally, the order of the blocks in logic programs must follow a flow like the one described below:

- IN: By-me objects in reading
- PROCESSING: object logic network
- OUT: writing in By-me

this diagram is given in all the examples in the manual and must be followed as a rule to avoid logics that cannot be correctly implemented by the Logic Unit.

3.12 Passing values between programs

Although each program is a logic network on its own, you can pass values between different programs using special logic blocks called "variables". To create a new variable you must do the following:

- Open the "LOGICHE" section of the main menu.
- Identify the section called "VARIABILI BINARIE" (if you want to create an ON/OFF type of variable) or "VARIABILI NUMERICHE".
- Press the corresponding "+" button and wait for the new variable to be added to the list.
- Select the new variable and drag it into the first program.

It is possible to assign a name to the variable in the "Proprietà generali" pane in order to identify it more easily within the programs in which it is to be used.

If you want to give the variable the value of an output node of a block (whether a logic or By-me block), simply connect it to the input node (left side) of the variable; vice versa, to use this value in other programs, connect the output node (right side) to the input node of another block (also in this case, either a logic or By-me block) as shown in the following figures.

- **You are advised to limit the use of variables used to transport only information retrieved from a logic network from one logic program to another.**
- **Use with caution: using the variables to "transport" data from By-me objects can lead to incorrect logic writing.**
- **IT IS NOT PERMITTED to create programs in which By-me blocks are in a different position other than IN and OUT in a logic.**
- **The same By-me block can be involved in more than one program as input but only in one program as output; this is to avoid malfunctions.**

Changing the name of a variable
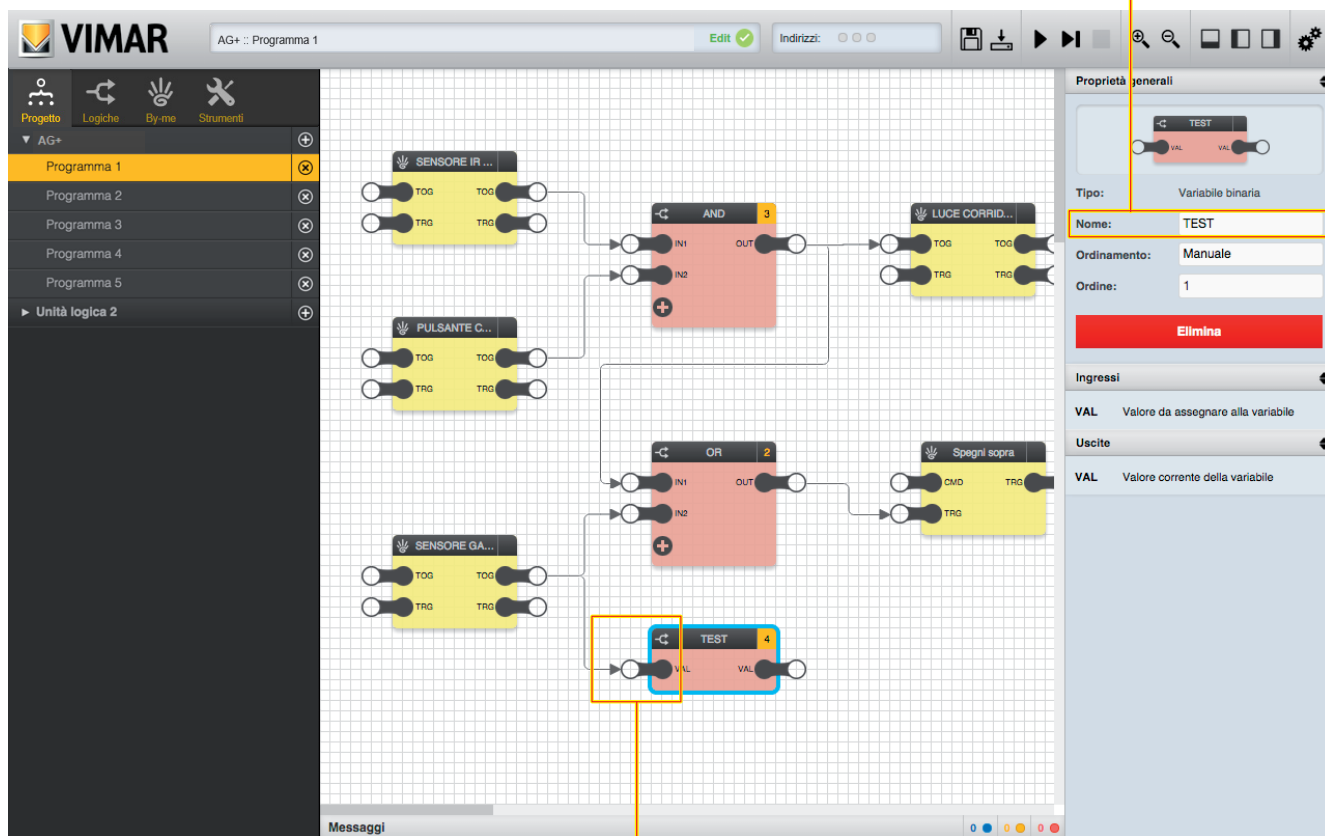
Assigning the value to a variable

Logic program editor

Logic programs

Using the value of a variable
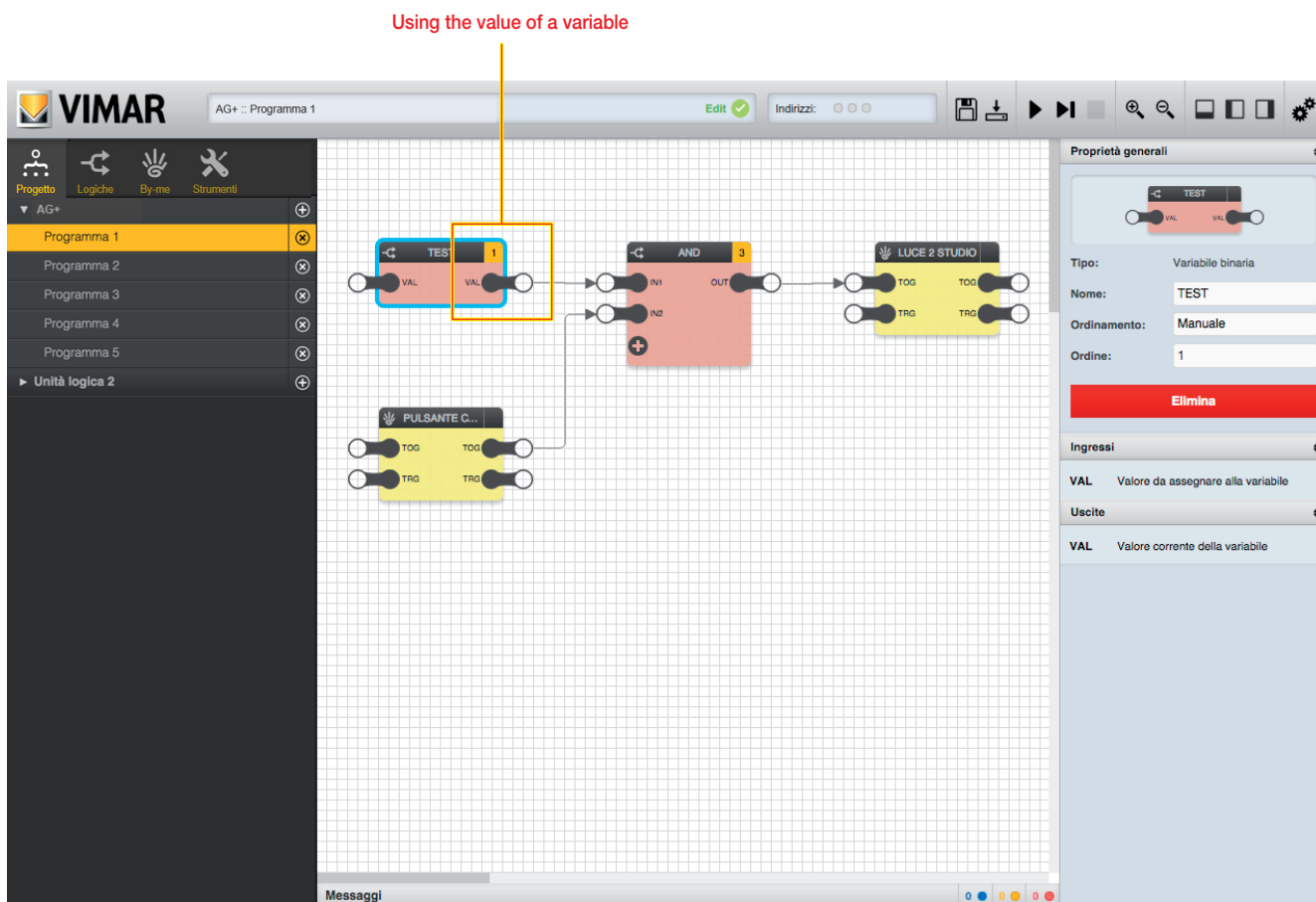
3.13 Data types

The input and output nodes of the blocks may include two types of data:

- BINARIO (BINARY): only the values 1 (ON) and 0 (OFF) are allowed.
- NUMERICO (NUMERICAL): any numerical value is allowed, possibly with specific restrictions depending on the block

These two data types are incompatible, so the editor prevents connecting binary nodes with numerical nodes and vice versa: as soon as you start dragging and dropping to create a connection, the incompatible nodes are made semi-transparent and cannot be released for creating the connection.

3.14 Saving

On closing the editor, the logic programs are automatically saved within the project so that they can be edited later.

You can however manually save the state of the logic programs – of all the logic units that there may be in the project – with the "SAVE" button on the toolbar; a progress screen is displayed while saving, and it is not possible to work on the logic programs.

3.15 Simulation

Before transferring the programs to the logic units, it is advisable to test them in the editor using "SIMULATION", which lets you manually enter the possible values received from the bus and check the behaviour of the logic networks, both continuously (repeated execution of the logic in real-time) and "step by step" (i.e. performing one calculation cycle at a time).

For more details on simulation, see chapter 7.

VIMAR

4. By-me Plus

4.1   Introduction

As mentioned previously, the By-me blocks enable reading values from the home automation bus and sending commands to the By-me groups after the logic processing carried out in the programs that contain them.

The By-me blocks available in the relevant section of the main menu are generated by a project import procedure, called up automatically when entering the editor.

> **CAUTION: If the timer parameter on the single functional unit is set (control or relay) in art. 01480, 01481, 01482, 01485 01486 and 01487, the Editor of the logic unit will not import the relative group.**

4.2 By-me blocks

4.2.1 Layout

As mentioned earlier, the By-me blocks appear graphically as shown in the following example:
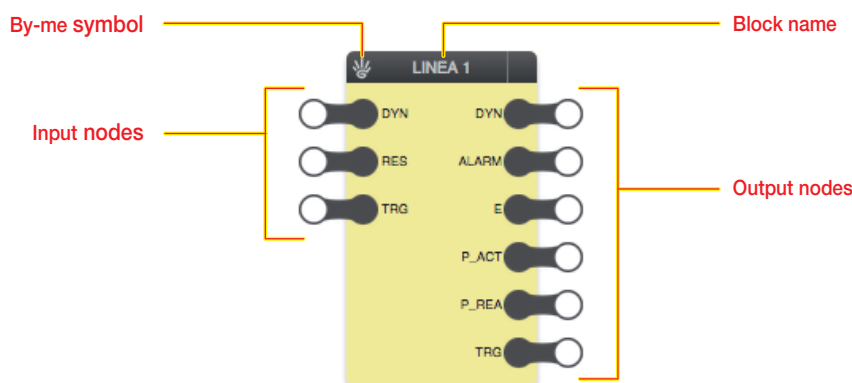
The By-me blocks are characterized by a yellow background.

4.2.2 Input nodes

The input nodes enable sending commands over the bus after the processing carried out in the logic programs; the available nodes depend on the type of By-me group, as detailed later on in this chapter.

The following options can be set by selecting a node and opening the "Proprietà generali" pane:

Control strategy	This determines with which criterion the node value is sent over the bus. Possible values: • **On changing**: the value is sent when it changes (unless the By-me block *trigger* is explicitly set to 1, as detailed below) • **Periodic sending**: the value is sent, as well as on changing, also periodically, with a settable time
Time for periodic sending	In the case of periodic sending, it determines the time between one send and the next one *Possible values:* 1 … 600 (seconds) *Note: low cyclical sending times can generate excess traffic on the bus.*
Initial Sync	Allows you to "force" sending the value of the node over the bus at start-up. For full details see section 4.2.2.1 on the next page.

> **CAUTION: Periodic sending can create traffic problems on the bus, particularly when they use low period values. This option must therefore be used only when it is strictly necessary to continuously reassert a value on the bus.**

In addition to the above options, the "Proprietà generali" pane also shows the possible values that the node can adopt; for binary nodes, the possible values are 0 (OFF) or 1 (ON), while for numerical nodes, the possible values depend on the type of node, and may have specific restrictions.

4.2.2.1 Initial Sync

The Sync function, which involves only the "actual" input nodes and not the TRG nodes, lets you force sending the value of the node over the bus on starting the logic unit (by default this feature is disabled).

If the flag ✓ is added for a given node, on starting up the logic unit will send a message over the bus with the value of the corresponding data point even if the latter has not changed compared to its default value.

This option, especially if extended to all the logic nodes, can generate higher traffic on the bus; due to this it should then only be used for nodes where it is necessary to immediately restore a value consistent with the logic gates (eg if you restart the system or logic unit after a power failure).

Caution: **The Sync function should never be selected for the input nodes of the By-me blocks used as "logic inputs"** (see the following figures).

Logic program editor

By-me Plus

Correct selection

Incorrect selection
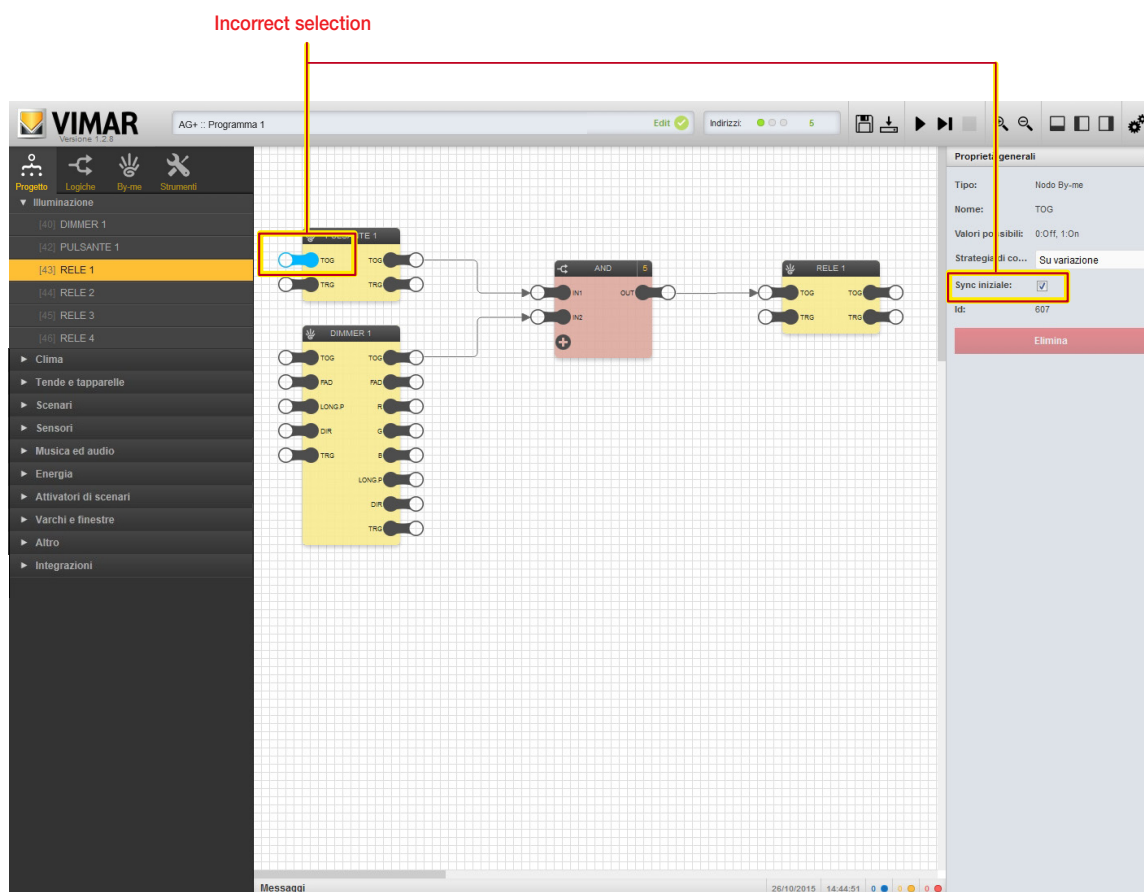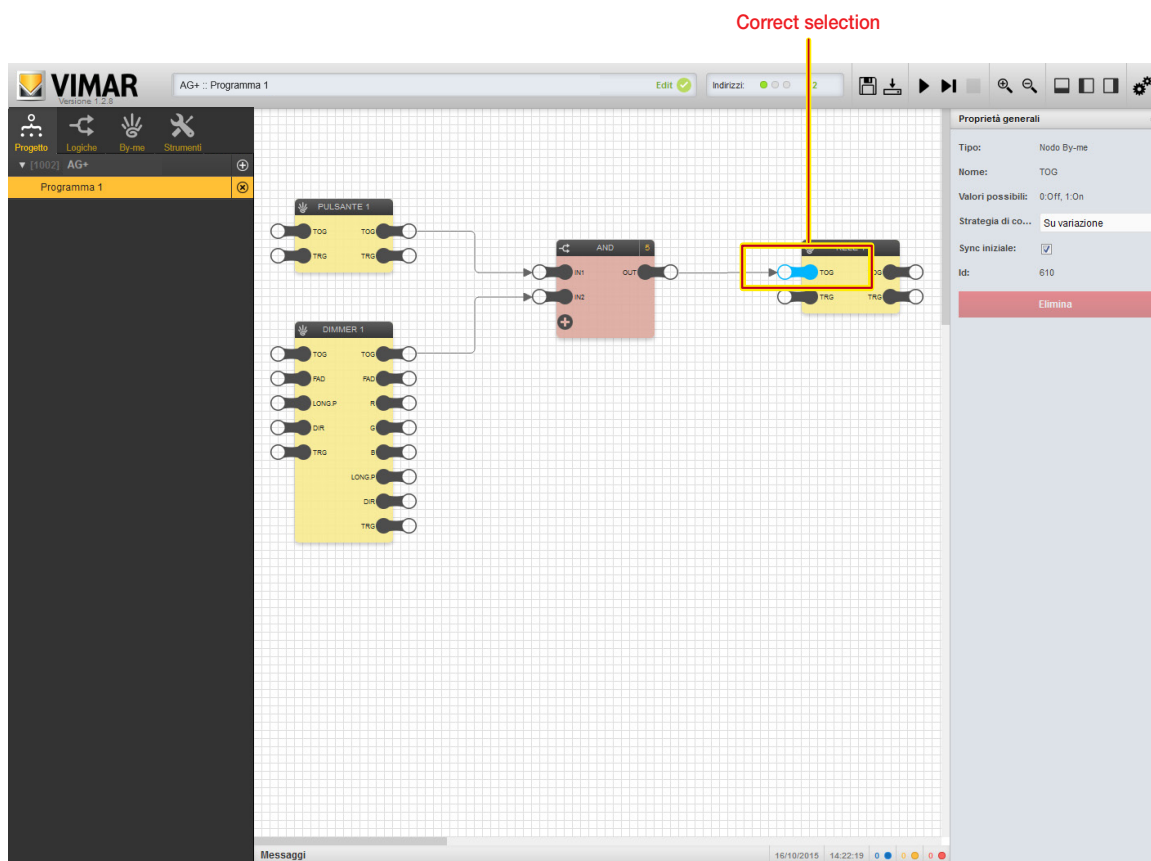
4.2.3 Output nodes

The output nodes enable receiving states from the bus and using them in the logic programs; the available nodes depend on the type of By-me group, as detailed later on in this chapter.

The following options can be set by selecting a node and opening the "Proprietà generali" pane:

Strategia all'avvio (Start-up strategy)	This determines which value the node must take on starting the logic unit. Possible values: • **Valore di default**: the "default value" set by the user is used (see below) • **Ultimo valore**: the last value received before switching off the logic unit is used • **Lettura da bus**: a request to read the status is sent to the device
Valore di default	This sets the default value of the node, which is used in the logic elements until a different value is received
Abilita polling	Enables periodic reading of the node value by polling the device over the bus
Tempo di polling	Periodic device polling time. *Possible values:* 1 … 600 (seconds) *Note: low polling times can generate excess traffic on the bus*

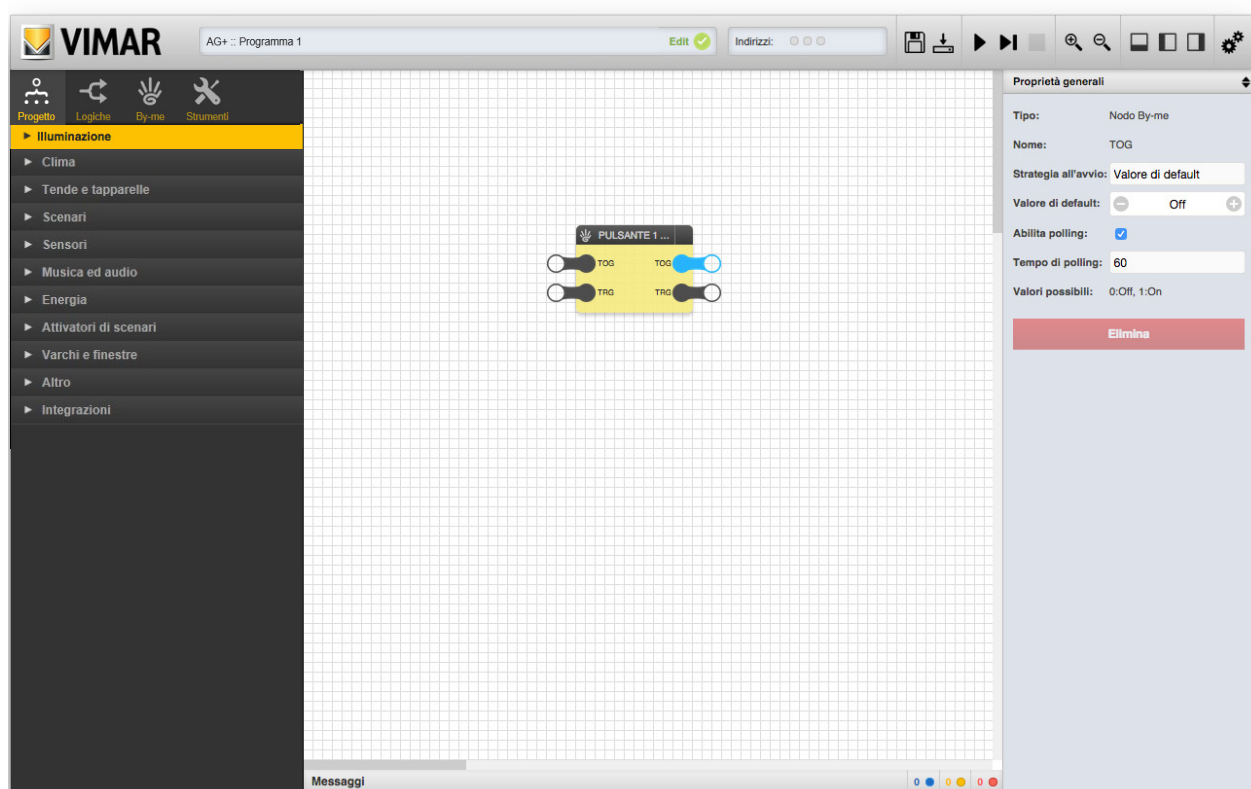| |
|---|
| CAUTION: Periodic sending can create traffic problems on the bus, particularly when they use low period values. This option must therefore be used only when it is strictly necessary to continuously reassert a value on the bus. |

4.2.4 Trigger

As mentioned in chapter 4, the By-me blocks have two trigger nodes, one at input and one at output.

The input trigger makes it possible to force sending commands related to the input nodes (connected to other blocks), even though their value has not changed. When this node is set to 1 (through a connection starting from a logic block n the program), the logic unit sends the commands over the bus, regardless of the current value and of any periodic sending; to repeat forced sending, it is necessary to set the trigger to 0 and then back to 1.

The trigger output, on the other hand, is set to 1 by the logic unit whenever data are received from the bus on one of the output nodes (connected to other blocks), even if the value has not changed; the trigger remains on 1 for the cycle and then goes back to 0, until data are next received from the bus.

Using the "Nodi coinvolti" option in the "Proprietà generali" pane, for both triggers, it is possible to set which nodes in the By-me block trigger the output trigger signal, or cause telegrams to be sent to the corresponding group addresses on the bus for each of the input trigger.

By-me Plus
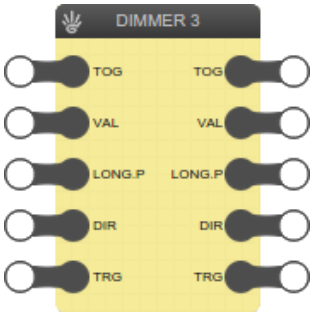
> CAUTION: The images shown for the various By-me blocks are the most representative ones. They are not intended to be complete and comprehensive as the shape and the presence of nodes depends on the configuration and the type of devices in the By-me group.

4.3 Lighting

4.3.1 Luci ON/OFF (Lights ON/OFF)

Preview:					
Nodes:	**TAG**	**Description**	**TYPE**	**IN**	**OUT**
	TOG	On/Off control *Possible values:* 0 ➜ OFF 1 ➜ ON	S	•	•
	TRG	Trigger for sending/receiving from/to bus	T	•	•

4.3.2 Dimmer

Preview:					
Nodes:	**TAG**	**Description**	**TYPE**	**IN**	**OUT**
	TOG	On/Off control *Possible values:* 0 ➜ OFF 1 ➜ ON	S	•	•
	VAL	Percentage adjustment *Possible values:* 0 … 100 [%]	S	•	•
	LONG.P	Long press start/end *Possible values:* 0 ➜ STOP long press end 1 ➜ START long press start	S	•	•
	DIR	Long press direction *Possible values:* 0 ➜ Down 1 ➜ Up	S	•	•
	TRG	Trigger for sending/receiving from/to bus	T	•	•

4.3.3 RGB

Preview:					
Nodes:	**TAG**	**Description**	**TYPE**	**IN**	**OUT**
	TOG	On/Off control — *Possible values:* 0 ➜ OFF / 1 ➜ ON	S	•	•
	FAD	Fading show ON/OFF — *Possible values:* 0 ➜ OFF / 1 ➜ ON	S	•	•
	R	Red intensity — *Possible values:* 0 … 255 [%], 255=100%	S		•
	G	Green intensity — *Possible values:* 0 … 255 [%], 255=100%	S		•
	B	Blue intensity — *Possible values:* 0 … 255 [%], 255=100%	S		•
	LONG.P	Long press start/end — *Possible values:* 0 ➜ STOP long press end / 1 ➜ START long press start	S	•	•
	DIR	Long press direction — *Possible values:* 0 ➜ Down / 1 ➜ Up	S	•	•
	TRG	Trigger for sending/receiving from/to bus	T	•	•

4.3.4 Attuatore con uscita analogica proporzionale (Proportional analogue output actuator)

For example, the group must contain a device such as: Proportional analogue 4-output actuator art. 01466.

Preview:					
Nodes:	**TAG**	**Description**	**TYPE**	**IN**	**OUT**
	TOG	On/Off control — *Possible values:* 0 ➜ OFF / 1 ➜ ON	S	•	•
	VAL	Percentage value — *Possible values:* 0 … 100 [%]	S	•	•
	ALARM	Alarm — *Possible values:* 0 ➜ OFF / 1 ➜ ON — Must be set to 1 when the VAL input value exceeds a threshold	S		•
	TRG	Trigger for sending/receiving from/to bus	T	•	•

4.3.5 Regolazione luminosità (Brightness control)

For example, the group must contain a device such as: Device with 3 analogue signal inputs art.01467 (connecting to brightness sensor art. 01530).

Preview:	

Nodes:	TAG	Description		TYPE	IN	OUT
	TOG	On/Off control	*Possible values:* 0 ➜ OFF 1 ➜ ON	S	•	•
	VAL	Value	*Possible values:* 0 … 100 [%]	S	•	•
	ENABLE	Regulation sensor enable continuous brightness	*Possible values:* 0 ➜ OFF 1 ➜ ON	S	•	
	LONG.P	Long press start/end	*Possible values:* 0 ➜ STOP long press end 1 ➜ START long press start	S	•	•
	DIR	Long press direction	*Possible values:* 0 ➜ Down 1 ➜ Up	S	•	•
	TRG	Trigger for sending/receiving from/to bus		T	•	•

4.4 Roller shutters
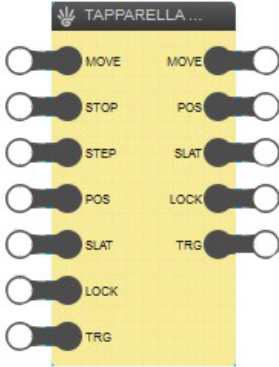
4.4.1 Tapparelle SU/GIU (Roller shutters UP/DOWN)

Preview:	

Nodes:	TAG	Description		TYPE	IN	OUT
	MOVE	Up/down movement	*Possible values:* OFF, 0 ➜ Up ON, 1 ➜ Down	S	•	•
	STOP	Stop movement	*Possible values:* ON ➜ Stop	S	•	•
	TRG	Trigger for sending/receiving from/to bus		T	•	•

4.4.2 Tapparelle lamellari (Lath roller shutters)

Preview:	

Nodes:	TAG	Description		TYPE	IN	OUT
	MOVE	Up/down movement	*Possible values:* 0 ➜ Up 1 ➜ Down	S	•	•
	STOP	Stop movement	*Possible values:* ON ➜ Stop	S	•	•
	STEP	Slat up/down adjustment	*Possible values:* 0 ➜ Up 1 ➜ Down	S	•	
	TRG	Trigger for sending/receiving from/to bus		T	•	•

4.4.3 Tapparelle lamellari con posizione (Lath roller shutters with position)

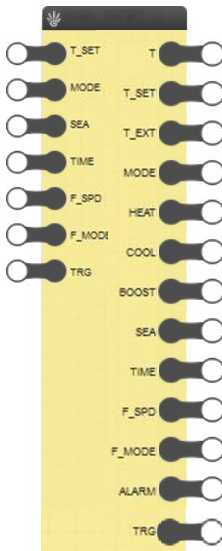| Preview:					
Nodes:	**TAG**	**Description**	**TYPE**	**IN**	**OUT**
	MOVE	Up/down movement — *Possible values:* 0 ➜ Up / 1 ➜ Down	S	•	•
	STOP	Stop movement — *Possible values:* ON ➜ Stop	S	•	
	STEP	Slat up/down adjustment — *Possible values:* 0 ➜ Up / 1 ➜ Down	S	•	
	POS	Percentage position — *Possible values:* 0 … 100 [%] / 0 =open, 100=closed	S	•	•
	SLAT	Percentage slat position — *Possible values:* 0 … 100 [%] / 0 =open, 100%=closed	S	•	•
	LOCK	Roller shutter lock — *Possible values:* ON, 1 ➜ Locked / OFF, 0 ➜ Unlocked	S	•	•
	TRG	Trigger for sending/receiving from/to bus	T	•	•

4.5 Climate control

4.5.1 Termostato/Sonda di temperatura (Thermostat/Temperature probe)

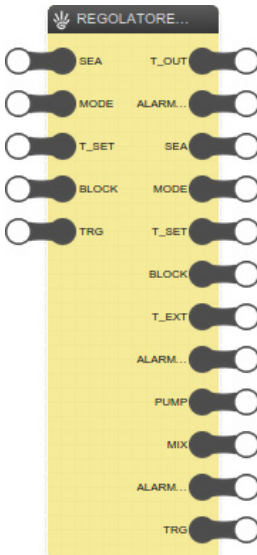N.B. The logic can manage only thermostats art. 02951.

Preview:	

Nodes:	TAG	Description	TYPE	IN	OUT
	T_SET	Temperature setpoint *Possible values:* 0 … 50 The setpoint refers to the current operating mode of the thermostat; by setting a value on this node, we therefore modify the On mode setpoint	S	•	•
	MODE	Operating mode *Possible values:* 0 ➜ Automatic 1 ➜ Manual 2 ➜ Reduction 3 ➜ Absent 4 ➜ Protection 5 ➜ Timed manual 6 ➜ OFF	S	•	•
	SEA	Season (control mode) *Possible values:* 0 ➜ Neutral zone 1 ➜ Air conditioning 2 ➜ Heating	S	•	•
	TIME	Timer *Possible values:* 0 … 255 [min] When set, it represents the time during which the thermostat stays in "Timed manual" mode (thus running with the fixed setpoint and ignoring any weekly programming) before returning to automatic mode. This parameter defines only the duration of this operating mode but does not determine the passage; the mode is determined by the value in the MODE node.	S	•	•
	HUM	Humidity sensor *Possible values:* 0 … 100 [%]	S		•
	F_SPD	Fan coil speed *Possible values:* 0 … 100 [%] Expressed as a percentage even with fan coils controlled on 3 ON-OFF speeds; in this case the 3 speeds correspond to the values 33%, 66% and 100%	S	•	•
	F_MODE	Fan coil mode *Possible values:* 0 ➜ Automatic 1 ➜ Manual	S	•	•
	T	Measured temperature *Possible values:* 0...40.0 [°C]	S		•
	T_EXT	Measured temperature (external probe) *Possible values:* -20...80.0 [°C]	S		•
	HEAT	Heating main output status *Possible values:* 0 ➜ OFF 1 ➜ ON	S		•
	COOL	Air conditioning main output status *Possible values:* 0 ➜ OFF 1 ➜ ON	S		•
	BOOST	Boost status (auxiliary heating/cooling) *Possible values:* 0 ➜ OFF 1 ➜ ON	S		•
	ALARM	Screed alarm *Possible values:* 0 ➜ OFF 1 ➜ ON	S		•
	TRG	Trigger for sending/receiving from/to bus	T	•	•

Note: The number and the type of nodes may depend on the specific configuration of the project

4.5.2 Regolatore climatico (HVAC controller)

Preview:	

Nodes:	TAG	Description		TYPE	IN	OUT
	SEA	Season (regulating mode)	*Possible values:* 0 ➜ Air conditioning 1 ➜ Heating	S	•	•
	MODE	Operation	*Possible values:* 0 ➜ Auto 1 ➜ Comfort 2 ➜ Economy 3 ➜ OFF	S	•	•
	T_SET	Setpoint	*Possible values:* 10 … 100 [°C]	S	•	•
	BLOCK	Controller block ON/OFF	*Possible values:* 0 ➜ No alarm 1 ➜ Alarm	S	•	•
	T_OUT	Delivery probe temperature	*Possible values:* -20 … 110 [°C]	S		•
	AL.T_OUT	Delivery probe alarm	*Possible values:* 0 ➜ No alarm 1 ➜ Alarm	S		•
	T_EXT	External temperature probe	*Possible values:* -20 … 70 [°C]	S		•
	AL.T_EXT	External probe alarm	*Possible values:* 0 ➜ No alarm 1 ➜ Alarm	S		•
	PUMP	Pump open/close	*Possible values:* 0 ➜ OFF 1 ➜ ON	S		•
	MIX	Valve opening	*Possible values:* 0 … 100 [%]	S		•
	AL.PROP	Proportional output alarm	*Possible values:* 0 ➜ No alarm 1 ➜ Alarm	S		•
	TRG	Trigger for sending/receiving from/to bus		T	•	•

Note: The number and the type of nodes may depend on the specific configuration of the project

Logic program editor

By-me Plus

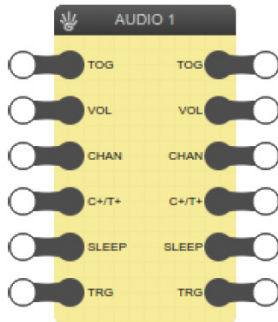### 4.6 Scenarios

4.6.1 Scenari By-me (By-me scenarios)

Preview:					
		Spegni tutto TRG TRG			

Nodes:	TAG	Description	TYPE	IN	OUT
	TRG	Trigger for sending/receiving from/to bus The input trigger is used to control the scenario; the output trigger notifies that the scenario has been run on the bus	T	•	•

CAUTION: It is not possible to create a logic in which groups interact with Scenarios that contain the same groups.
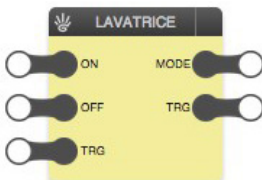
4.7 Audio

4.7.1 Zone di diffusione sonora (Speaker systems zones)

Preview:					
		AUDIO 1 TOG TOG VOL VOL CHAN CHAN C+/T+ C+/T+ SLEEP SLEEP TRG TRG			

Nodes:	TAG	Description		TYPE	IN	OUT
	TOG	On/Off control	Possible values: 0 ➜ OFF 1 ➜ ON	S	•	•
	VOL	Volume	Possible values: 0 … 99 [%]	S	•	•
	CHAN	Channel (selecting the sound source from a maximum of 4 available) Possible values: 1 … 4		S	•	•
	C+/T+	Channel+/Track+	Possible values: 0 ➜ Track+ 1 ➜ Channel+	S	•	•
	SLEEP	Timed ON/OFF switch	Possible values: 0 ➜ OFF 1 ➜ ON	S	•	•
	TRG	Trigger for sending/receiving from/to bus		T	•	•

4.8 Energy management

4.8.1 Carichi (Loads)

Preview:					
		LAVATRICE ON MODE OFF TRG TRG			

Nodes:	TAG	Description		TYPE	IN	OUT
	ON	Forcing ON	Possible values: 0 ➜ OFF 1 ➜ ON	S	•	
	OFF	Forcing OFF	Possible values: 0 ➜ OFF 1 ➜ ON	S	•	
	MODE	Operating mode	Possible values: 0 ➜ Automatic ON 1 ➜ Automatic OFF 2 ➜ Forced ON 3 ➜ Forced OFF	S		•
	TRG	Trigger for sending/receiving from/to bus		T	•	•

4.8.2 Gestori di linea (Line controllers)

This object represents part of the device 01455 linked to a single line in the system. There will therefore be as many "Line Manager" objects as there are lines configured in the system. According to this configuration, each line will measure (or not) consumption or production. See the notes concerning individual nodes for more information.

Preview:	

Nodes:	TAG	Description		TYPE	IN	OUT
	DYN	Dynamic mode *Possible values:* 0 → OFF / 1 → ON Dynamic Mode of meters. Present in the line manager if the system configuration allows a meter in the line manager index. This parameter is useful when ON; the active power measurement is sent to the device for a time equal to the parameter "Measurement refresh duration" and a frequency equal to the "Frequenza rinfresco misura" (Measurement refresh frequency) parameter. It can be used by a display device to show the value in real time, such as for example on opening a page in the touch screen; after the set time the transmission ends.		S	•	•
	RES	Partial reset *Possible values:* 0... 429496729 [Wh] / or -2147483647/+2147483647 [Wh] Used to set the partial energy measurement of the meter in the line manager index to a given value; it is present if the system configuration allows a meter in the line manager index. This parameter forces the partial energy value to the value set here and is useful for aligning the energy value calculated by the Vimar device to that of an external meter. The measurement MUST be of the same time, which depends on how the system has been configured and how the current sensors are positioned: exchanged (if for production), produced (photovoltaic meter) or consumer (without production).		S	•	
	ALARM	At least one load on the line is off *Possible values:* 0 → OFF / 1 → ON		S		•
	T_MIN (*)	Minimum threshold *Possible values:* -2147483647/+2147483647 [kW] It represents the minimum energy threshold for the load control logic. Threshold value 1 refers to the value set in the line manager device in question.		S		•
	T_MAX (*)	Maximum threshold *Possible values:* -2147483647/+2147483647 [kW] It represents the maximum energy threshold for the load control logic. Threshold value 2 refers to the value set in the line manager device in question.		S		•
	E (*)	Partial energy *Possible values:* -2147483647/+2147483647 [Wh] It represents the energy measured from the last reset.		S		•
	P_ACT (*)	Active power *Possible values:* -2147483647/+2147483647 [kW] It represents the measured power. It refers to the active power read by the meter in the line manager. Depending on the system configuration this power can have different meanings (see table on the next page).		S		•
	P_REA (*)	Reactive power *Possible values:* -2147483647/+2147483647 [kVAR] It represents the reactive quota of the measured power. It refers to the reactive power read by the meter in the line manager.		S		•
	TRG	Trigger for sending/receiving from/to bus		T	•	•

(*) Some nodes might not be present depending on the configuration of the system. In particular, the energy and power data are available only when an external meter is associated to the line.
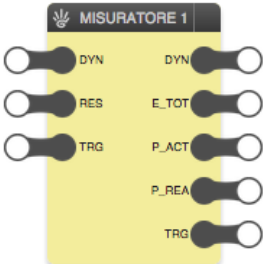
Note 1: The P_ACT values depend on the type of system: max current supported by the cable on which the measurement is taken and the power output from the energy distributor. For example, in a domestic system with a standard Enel contract, this can reach 3.3kW.

Note 2: The P_REA values depend upon the inductive/capacitive absorption characteristics of the appliances in the system.

4.8.3 Misuratori (Meters)

Like the previous block, this object represents part of the device 01455 linked to a single meter in the system.

Preview:	

Nodes:	TAG	Description	TYPE	IN	OUT
	DYN	Dynamic mode — *Possible values:* 0 ➔ OFF / 1 ➔ ON Dynamic Mode of meters. Present in the line manager if the system configuration allows a meter in the line manager index. This parameter is useful when ON; the active power measurement is sent to the device for a time equal to the parameter "Measurement refresh duration" and a frequency equal to the "Frequenza rinfresco misura" (Measurement refresh frequency) parameter. It can be used by a display device to show the value in real time, such as for example on opening a page in the touch screen; after the set time the transmission ends.	S	•	•
	RES	Partial reset — *Possible values: -2147483647/+2147483647 [Wh]* Used to set the partial energy measurement of the meter in the line manager index to a given value; it is present if the system configuration allows a meter in the line manager index. This parameter forces the partial energy value to the value set here and is useful for aligning the energy value calculated by the Vimar device to that of an external meter. The measurement MUST be of the same time, which depends on how the system has been configured and how the current sensors are positioned: exchanged (if for production), produced (photovoltaic meter) or consumer (without production).	S	•	
	E_TOT	Total energy — *Possible values: -2147483647/+2147483647 [Wh]* It represents the overall energy measured; for plants with production, it is the difference between energy drawn off (consumed) and fed in (produced)	S		•
	E_IN (*)	Energy drawn (from the mains) — *Possible values: -2147483647/+2147483647 [Wh]* It represents the total energy consumed, whatever the production			•
	E_OUT (*)	Energy fed (into the mains) — *Possible values: -2147483647/+2147483647 [Wh]* It represents the total energy produced (if production is available), whatever the consumption	S		•
	P_ACT	Active power — *Possible values: -2147483647/+2147483647 [kW]* It represents the measured power. It refers to the active power read by the meter in the line manager. Depending on the system configuration this power can have different meanings (see table below).	S		•
	P_REA	Reactive power — *Possible values: -2147483647/+2147483647 [kVAR]* It represents the reactive component of the measured power.	S		•
	TRG	Trigger for sending/receiving from/to bus	T	•	•

(*) Some nodes might not be available depending on the firmware version of the device

Note 1: The P_ACT values depend on the type of system: max current supported by the cable on which the measurement is taken and the power output from the energy distributor. For example, in a domestic system with a standard Enel contract, this can reach 3.3 kW.

Note 2: The P_REA values depend upon the inductive/capacitive absorption characteristics of the appliances in the system.

EXAMPLES OF USE OF THE ENERGY MANAGEMENT FUNCTIONS BY MEANS OF THE LOGIC UNIT

Single-phase system with no generation Refer to the By-me Plus manual for the diagram.	Load control module 01455 BLOCK By-me Meter 1: Node P_ACT = Power Exchanged Note: > 0 = Drawn off; < 0 Fed in The By-me block Line 1 is not used for metering.
Single-phase system with "local" generation Refer to the By-me Plus manual for the diagram.	Load control module 01455 BLOCK By-me Meter 1: Node P_ACT = Power Exchanged Note: > 0 = Drawn off; < 0 Fed in BLOCK By-me Meter 2: Node P_ACT = Power Generated Note: must be ≥ 0 The By-me block Line 1 and Line 2 are not used for metering.
Single-phase system with "remote" generation Refer to the By-me Plus manual for the diagram.	Load control module 01455 BLOCK By-me Line 1: Node P_ACT = Power Generated Note: must be ≥ 0 BLOCK By-me Meter 1: Node P_ACT = Power Exchanged Note: > 0 = Drawn off; < 0 Fed in Power meter 01450 BLOCK By-me Meter 1: Node P_ACT = Power Generated Note: must be ≥ 0 This is the same as indicated above, use one block or the other according to the needs of the logic program
Three-phase system without production Refer to the By-me Plus manual for the diagram.	Load control module 01455 BLOCK By-me Meter 1: Node P_ACT = Power Exchanged Note: > 0 = Drawn off; < 0 Fed in The By-me block Line 1 is not used for metering. BLOCK By-me Meter 2: Node P_ACT = Power Exchanged Note: > 0 = Drawn off; < 0 Fed in The By-me block Line 2 is not used for metering. BLOCK By-me Meter 3: Node P_ACT = Power Exchanged Note: > 0 = Drawn off; < 0 Fed in The By-me block Line 3 is not used for metering.
Three-phase system with generation (from one to three phases) Refer to the By-me Plus manual for the diagram.	Load control module 01455 BLOCK By-me Line 1-2-3: Node P_ACT = Power Generated Note: must be ≥ 0 BLOCK By-me Meter 1-2-3: Node P_ACT = Power Exchanged Note: > 0 = Drawn off; < 0 Fed in Power meter 01450 BLOCK By-me Meter 1-2-3: Node P_ACT = Power Generated Note: must be ≥ 0 This is the same as indicated above, use one block or the other according to the needs of the logic program

4.8.4 Contatori (Counters)

Preview:					
Nodes:	**TAG**	**Description**	**TYPE**	**IN**	**OUT**
	RES	Partial reset *Possible values:* 0... 4294967296	S	•	
	VAL	Meter (the description depends on the type of meter) *Possible values:* 0... 4294967296	S		•
	TRG	Trigger for sending/receiving from/to bus	T	•	•

Note: The number and the type of nodes may depend on the specific configuration of the project

4.9 Sensors

4.9.1 Sensori in sola visualizzazione (Display-only sensors)

Preview:					
Nodes:	**TAG**	**Description**	**TYPE**	**IN**	**OUT**
	VAL	Value *Possible values:* any numerical value	S		•
	ALARM	Alarm *Possible values:* 0 ➜ No alarm 1 ➜ Alarm	S		•
	TRG	Trigger for sending/receiving from/to bus	T		•

4.9.2 Sensori con controllo (Sensors with control)

Preview:					
Nodes:	**TAG**	**Description**	**TYPE**	**IN**	**OUT**
	ENABLE	Sensor activation *Possible values:* 0 ➜ OFF 1 ➜ ON	S	•	
	SET	Setpoint *Possible values:* any numerical value Used to set the threshold above which the output associated to the sensor is enabled	S	•	
	TOG	ON/OFF *Possible values:* 0 ➜ OFF 1 ➜ ON	S		•
	VAL	Value *Possible values:* any numerical value	S		•
	TRG	Trigger for sending/receiving from/to bus	T	•	•

4.9.3 Weather station

Preview:

Logic program editor

By-me Plus

VIMAR

Nodes:	TAG	Description		TYPE	IN	OUT
	NIGHT	Day/Night	*Possible values:* 0 ➜ OFF (day) 1 ➜ ON (night)	S		•
	RAIN	Rain/No rain	*Possible values:* 0 ➜ OFF (no rain) 1 ➜ ON (rain)	S		•
	T	Temperature	*Possible values:* -273°C...670760 °C	S		•
	T_MIN	Minimum temperature measured	*Possible values:* -273°C...670760 °C	S		•
	T_MAX	Maximum temperature measured	*Possible values:* -273°C...670760 °C	S		•
	T.REQ	Minimum/maximum temperature request	*Possible values:* 0 ➜ OFF 1 ➜ ON	S	•	•
	T.RES	Reset temperature	*Possible values:* -273°C...670760 °C	S	•	•
	AL.T	Temperature sensor fault	*Possible values:* 0 ➜ No alarm 1 ➜ Alarm	S		•
	T.TSET	Temperature setpoint to reach	*Possible values:* -273°C...670760 °C	S	•	•
	T.ASET	Current temperature setpoint	*Possible values:* -273°C...670760 °C	S		•
	T.ESET	Temperature setpoint exceeded	*Possible values:* 0 ➜ OFF 1 ➜ ON	S		•
	W	Wind speed	*Possible values:* 0 m/s...670760 m/s	S		•
	W_MAX	Maximum wind speed	*Possible values:* 0 m/s...670760 m/s	S		•
	W.RES	Reset maximum wind speed	*Possible values:* 0 ➜ OFF 1 ➜ ON	S	•	•
	AL.W	Wind speed sensor fault	*Possible values:* 0 ➜ No alarm 1 ➜ Alarm	S		•
	W.TSET	Wind speed setpoint to reach	*Possible values:* 0 m/s...670760 m/s	S	•	•
	W.ASET	Current wind speed setpoint	*Possible values:* 0 m/s...670760 m/s	S		•
	W.ESET	Wind speed setpoint exceeded	*Possible values:* 0 ➜ OFF 1 ➜ ON	S		•
	BRI	Brightness	*Possible values:* 0 lx...670760 lx	S		•
	B.TSET	Brightness setpoint to reach	*Possible values:* 0 lx...670760 lx	S	•	•
	B.ASET	Current brightness setpoint	*Possible values:* 0 lx...670760 lx	S		•
	B.ESET	Brightness setpoint exceeded	*Possible values:* 0 ➜ OFF 1 ➜ ON	S		•
	D.TSET	Dawn setpoint to reach	*Possible values:* 0 lx...670760 lx	S	•	•
	D.ASET	Current dawn setpoint	*Possible values:* 0 lx...670760 lx	S		•
	D.ESET	Dawn setpoint exceeded	*Possible values:* 0 ➜ OFF 1 ➜ ON	S		•
	TRG	Trigger	*Possible values:* 0 ➜ OFF 1 ➜ ON	T	•	•

N.B.: the number of nodes and the related data type may depend on the configuration made.

Logic program editor

By-me Plus

4.10 KNX integration

Given the structural compatibility of By-me and KNX, you can create mixed systems.

Some control widgets for KNX device communication objects (configured via ETS) can be exported to By-me supervisors, such as touchscreens.
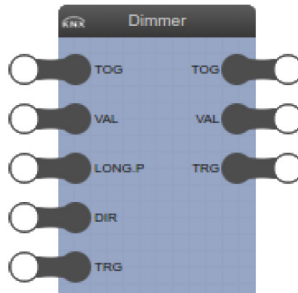
In addition, through the By-me logic unit, you can create integrated logic elements involving the devices of the two systems.

The following blocks illustrate how it is possible to achieve this integration by connecting nodes that allow sending/receiving commands or writing/reading values on datapoints shared between By-me and KNX installations.
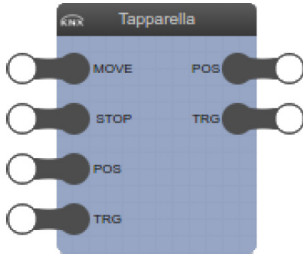
4.10.1 Attuatore (Actuator)

Preview:					
Nodes:	**TAG**	**Description**	**TYPE**	**IN**	**OUT**
	TOG	On/Off control _Possible values:_ 0 ➜ OFF / 1 ➜ ON	S	•	•
	TRG	Trigger for sending/receiving from/to bus	T	•	•
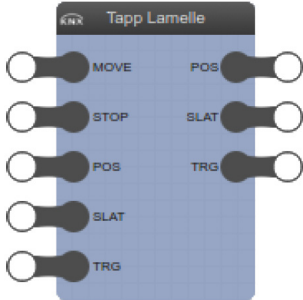
4.10.2 Dimmer

Preview:					
Nodes:	**TAG**	**Description**	**TYPE**	**IN**	**OUT**
	TOG	On/Off control _Possible values:_ 0 ➜ OFF / 1 ➜ ON	S	•	•
	VAL	Percentage adjustment _Possible values:_ 0 … 100 [%]	S	•	•
	LONG.P	Long press start/end _Possible values:_ 0 ➜ STOP long press end / 1 ➜ START long press start	S	•	•
	TRG	Trigger for sending/receiving from/to bus	T	•	•

4.10.3 Tapparelle SU/GIU (Roller shutters UP/DOWN)

Preview:	

Nodes:	TAG	Description			TYPE	IN	OUT
	MOVE	Up/down movement	Possible values:	0 → Up 1 → Down	S	•	
	STOP	Stop movement	Possible values:	1 → Stop	S	•	
	POS	Percentage position	Possible values:	0 … 100 [%]	S	•	•
	TRG	Trigger for sending/receiving from/to bus			T	•	•

4.10.4 Tapparelle lamellari (Lath roller shutters)
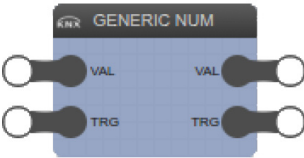
Preview:	

Nodes:	TAG	Description			TYPE	IN	OUT
	MOVE	Up/down movement	Possible values:	0 → Up 1 → Down	S	•	
	STOP	Stop movement	Possible values:	1 → Stop	S	•	
	STEP	Slat up/down adjustment	Possible values:	0 → Up 8 → Down	S	•	
	POS	Percentage position	Possible values:	0 … 100 [%]	S	•	•
	SLAT	Slat percentage position	Possible values:	0 … 100 [%]	S	•	•
	TRG	Trigger for sending/receiving from/to bus			T	•	•

4.10.5 Generic Boolean

Preview:	

Nodes:	TAG	Description			TYPE	IN	OUT
	TOG	Boolean value	Possible values:	0 → OFF 1 → ON	S	•	•
	TRG	Trigger for sending/receiving from/to bus			T	•	•

CAUTION: For the block to function correctly, it is necessary to set a default value that is consistent with the actual operation.

4.10.6 Generic Numerical

Preview:					

Nodes:	TAG	Description	TYPE	IN	OUT
	VAL	Numerical value *Possible values:* Variable range depending on the DPT represented	S	•	•
	TRG	Trigger for sending/receiving from/to bus	T	•	•

CAUTION:

• For the block to function correctly, it is necessary to set a default value that is consistent with the actual operation.

• For the "Generico booleano" and "Generico numerico" blocks to function correctly, the nodes must be connected to other nodes of the same type.

5. By-alarm and video door entry systems

5.1 By-alarm
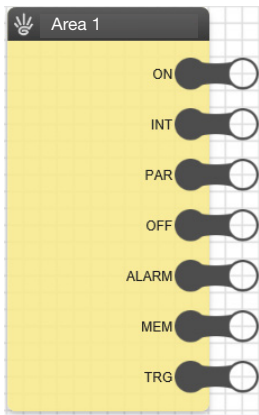
By-alarm blocks allow values to be read from the intrusion detection alarm system and send commands to the By-me groups after the logic processing carried out in the programs that contain them. These blocks are managed in the same way as By-me blocks, and are located in the "Integrazioni" application.

To access the logic blocks, it is first necessary to configure the integration and applications containing the integration blocks in the home automation system gateway.

Caution: The By-alarm blocks are always visible even if not available in your system.

5.1.1 Area
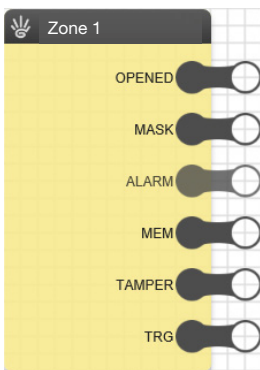
Preview:					
Nodes:	**TAG**	**Description**	**TYPE**	**IN**	**OUT**
	ON	Active area ON *Possible values:* 0...1 (0=OFF, 1=ON)	S		•
	INT	Active area INT *Possible values:* 0...1 (0=OFF, 1=ON)	S		•
	PAR	Active area PAR *Possible values:* 0...1 (0=OFF, 1=ON)	S		•
	OFF	Area OFF *Possible values:* 0...1 (0=OFF, 1=ON)	S		•
	ALARM	Area alarm *Possible values:* 0...1 (0=OFF, 1=ON)	S		•
	MEM	Area in alarm memory *Possible values:* 0...1 (0=OFF, 1=ON)	S		•
	TRG	Trigger for sending/receiving from/to bus	T		•

5.2.1 Zona
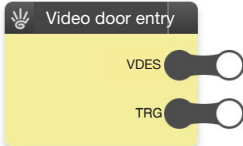
Preview:					
Nodes:	**TAG**	**Description**	**TYPE**	**IN**	**OUT**
	OPENED	Zone opened (e.g. magnetic contact) *Possible values:* 0...1 (0=OFF, 1=ON)	S		•
	MASK	Zone masking *Possible values:* 0...1 (0=OFF, 1=ON)	S		•
	ALARM	Zone alarm *Possible values:* 0...1 (0=OFF, 1=ON)	S		•
	MEM	Zone in alarm memory *Possible values:* 0...1 (0=OFF, 1=ON)	S		•
	TAMPER	Zone tamper alarm *Possible values:* 0...1 (0=OFF, 1=ON)	S		•
	TRG	Trigger for sending/receiving from/to bus	T		•

5.2 Video door entry system

The videocitofonia blocks can read the status of a video door entry system call and send it to perform an actuation command.

These blocks are also managed in the same way as By-me blocks, and are located in the "Integrazioni" application.

Preview:	
	☼ Video door entry VDES ●○ TRG ●○

Nodes:	TAG	Description		TYPE	IN	OUT
	VDES	Video door entry system status	*Possible values:* 0...1 (0=OFF, 1=ON)	S		•
	TRG	Trigger for sending/receiving from/to bus		T		•

6. Logic functions

6.1 Introduction

The logic blocks enable performing operations on one or more input values and they return one or more output values, which can be connected with other logic blocks or By-me blocks.

6.2 Logic blocks

6.2.1 Layout

As mentioned earlier, the logic blocks appear graphically as shown in the following example:
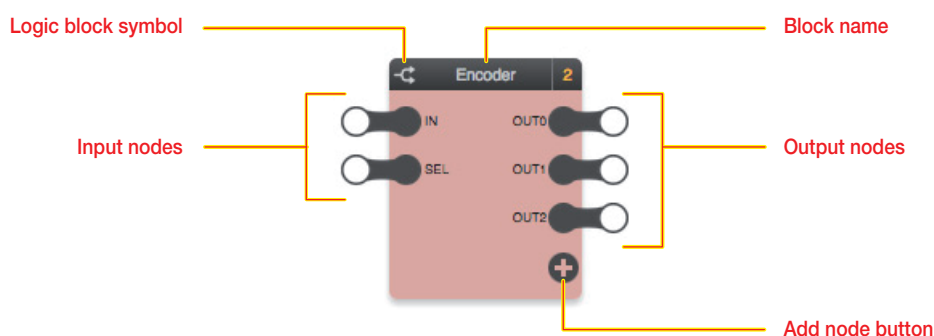
The logic blocks are characterized by an amber background.

6.2.2 Input nodes

The input nodes enable passing values to the logic functions. By selecting an input node and opening the "Proprietà generali" pane, you can set the following options:

Valore di default	This sets the value of the node to be used when starting to run, until a different value is received, or if the node is not connected to another block
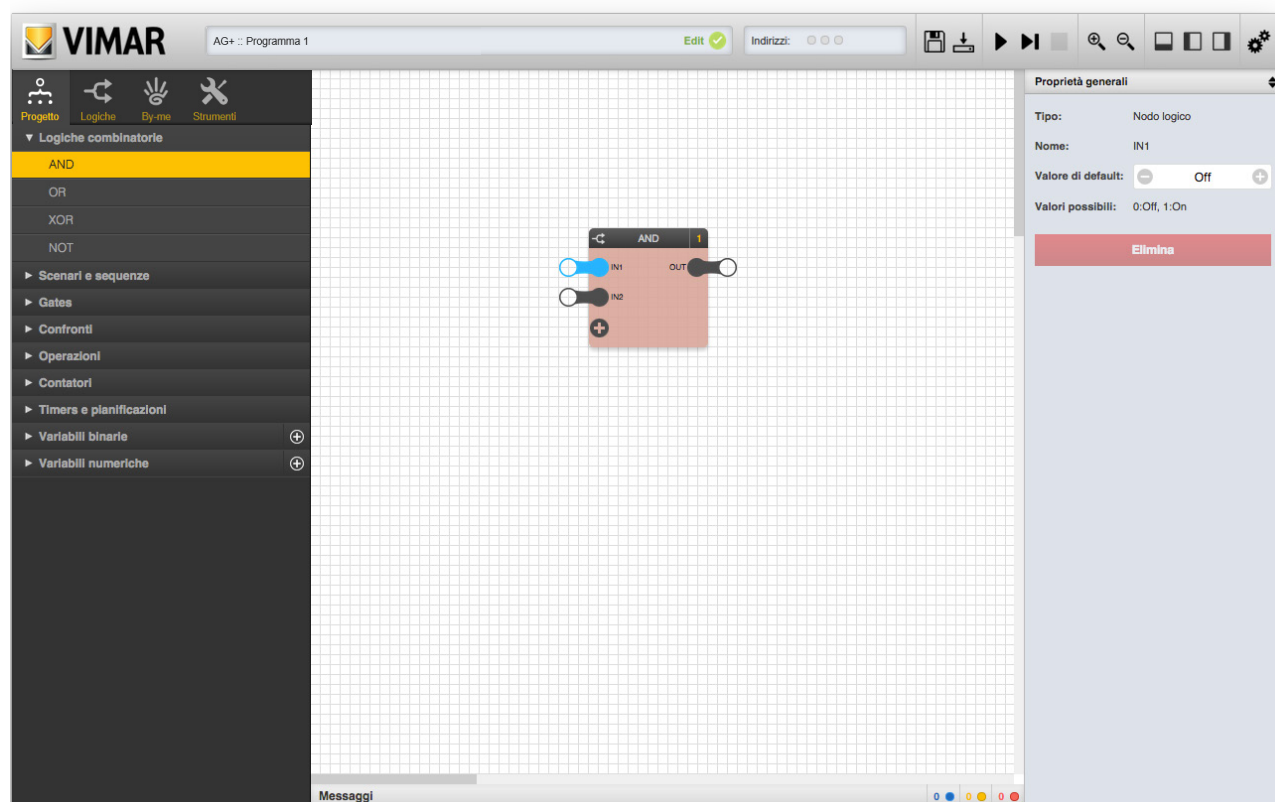
In addition to the above options, the "Proprietà generali" pane also shows the possible values that the node can adopt; for binary nodes, the possible values are 0 (OFF) or 1 (ON), while for numerical nodes, the possible values depend on the type of node, and may have specific restrictions.

Logic functions

6.2.3 Output nodes

The output nodes return the results of the logic function paired with the block and enable passing them to other blocks, of both the logic and By-me type.

There is no option for the output nodes of logic blocks.

6.2.4 Adding and removing nodes

Some blocks have a variable number of nodes; in these cases, the block, once dragged from the side menu, typically contains a minimal set of nodes, which may be increased up to a maximum number of nodes, by pressing the "+" button.

To remove a previously added node, perform the following steps:
- Select the node
- Open the "Proprietà generali" pane
- Press the "ELIMINA" button
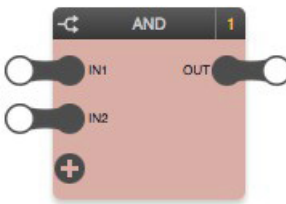
Any connections associated with the node will be deleted.

6.2.5 Types of blocks and nodes
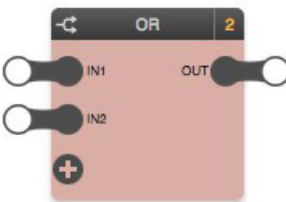
In some cases Logic Blocks (or nodes of Logic Blocks) are separated into "binary" and "numerical". The former are designed to mange boolean signals, i.e. can only have True/False (or "ON/OFF") values. The latter can hand numerical data. The Editor controls the correspondence of these types and prevents connections between nodes of different types.

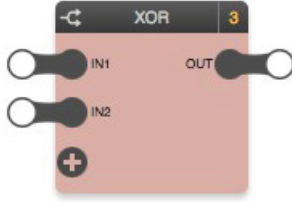## 6.3 Combinational logic gates

6.3.1 AND

Description:	Performs the AND logic function between two or more binary inputs (up to a maximum of 10)				
Preview:					

Nodes:	TAG	Description		TYPE	IN	OUT
	IN1 … IN10	Input 1 … 10	Possible values: 0 ➜ OFF 1 ➜ ON	M	•	
	OUT	Output	Possible values: 0 ➜ OFF 1 ➜ ON	M		•
	+	Add node			•	

6.3.2 OR
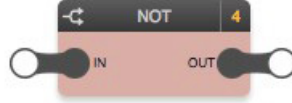
Description:	Performs the OR logic function between two or more binary inputs (up to a maximum of 10)				
Preview:					

Nodes:	TAG	Description		TYPE	IN	OUT
	IN1 … IN10	Input 1 … 10	Possible values: 0 ➜ OFF 1 ➜ ON	M	•	
	OUT	Output	Possible values: 0 ➜ OFF 1 ➜ ON	M		•
	+	Add node			•	

Logic program editor

Logic functions

6.3.3 XOR
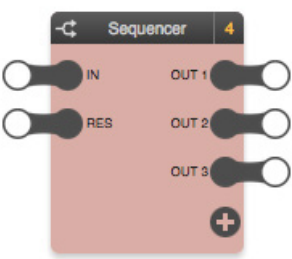
Description:	Performs the XOR logic function between two or more binary inputs (up to a maximum of 10)				
Preview:					

Nodes:	TAG	Description		TYPE	IN	OUT
	IN1 … IN10	Input 1 … 10	Possible values: 0 ➜ OFF 1 ➜ ON	M	•	
	OUT	Output	Possible values: 0 ➜ OFF 1 ➜ ON	M		•
	+	Add node			•	

6.3.4 NOT

Description:	Performs the NOT logic function of the input				
Preview:					

Nodes:	TAG	Description		TYPE	IN	OUT
	IN	Input	Possible values: 0 ➜ OFF 1 ➜ ON	M	•	
	OUT	Output	Possible values: 0 ➜ OFF 1 ➜ ON	M		•
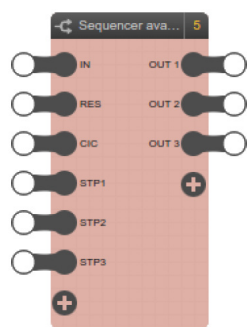
6.4 Scenarios and sequences

6.4.1 Sequencer

Description:	According to the input status IN, it toggles in sequence up to 10 boolean outputs, keeping each of them active for a settable time				
Preview:					

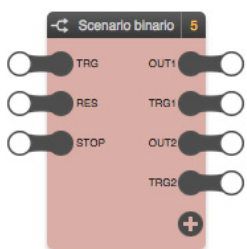| Nodes:	TAG	Description		TYPE	IN	OUT
	IN	Start sequence	Possible values: 0 ➜ OFF 1 ➜ ON	M	•	
	RES	Reset sequence	Possible values: 0 ➜ OFF 1 ➜ ON	M	•	
	OUT1 … OUT10	Output 1… 10	Possible values: 0 ➜ OFF 1 ➜ ON	M		•
	+	Add node				•
Options:	Sequenza ciclica	Determines whether the sequence should be repeated once completed Possible values: VERO/FALSO (TRUE/FALSE)				
	Durata passo (Step duration) 1 … 10	Waiting time between step X and the next	Possible values: from 1 second to 12 hours The step is 1 second and can be specified in the format HH:MM:SS (hours, minutes, seconds)			

Logic functions

6.4.2 Sequencer avanzato (Advanced Sequencer)

Description:	The function is similar to that of the Sequencer with the difference that some options have been converted into nodes so that they can be changed dynamically via the virtual datapoints
Preview:	

Nodes:	TAG	Description		TYPE	IN	OUT
	IN	Start sequence	Possible values: 0 ➜ OFF 1 ➜ ON	M	•	
	RES	Reset sequence	Possible values: 0 ➜ OFF 1 ➜ ON	M	•	
	CIC	Cyclic sequence	Possible values: 0 ➜ OFF 1 ➜ ON	S	•	
	STP1…STP10	Durata passo 1 ... 10	Possible values: any	S	•	
	OUT1…OUT10	Output 1 ... 10	Possible values: any	M		•
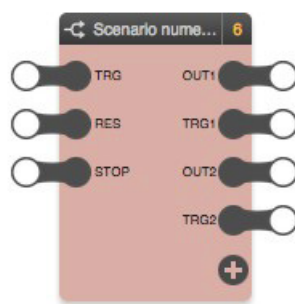
CAUTION:
The values to provide to the Advanced Sequencer logic unit to define the activation time for each of its outputs (via nodes STP1..STP10); It must always be expressed in seconds.

6.4.3 Scenario binario (Binary scenario)

Description:	When a pulse is received on the TRG input, it runs a sequence of boolean commands, each of which can be set, possibly spacing each command with a preset time, common to all the outputs
Preview:	

Nodes:	TAG	Description		TYPE	IN	OUT
	TRG	Input trigger	Possible values: 0 ➜ OFF 1 ➜ ON	T	•	
	RES	Reset scenario Returns all the scenario outputs to the initial status (default).	Possible values: 0 ➜ OFF 1 ➜ ON	M	•	
	STOP	Stop scenario If on, the scenario is stopped. The scenario is resumed when the STOP signal is disabled (useful above all when setting time > 0 as an interval between the activation of the scenario outputs).	Possible values: 0 ➜ OFF 1 ➜ ON	M	•	
	OUT1 … OUT10	Output 1… 10	Possible values: 0 ➜ OFF 1 ➜ ON	S		•
	TRG1 … TRG10	Trigger 1 … 10	Possible values: 0 ➜ OFF 1 ➜ ON	T		•
	+	Add node (and its trigger)				•
Options:	Intervallo uscite	Waiting time between output command	Possible values: 1 … 60 (seconds)			
	Set uscita 1 … 10	Value to set on output 1 … 10	Possible values: 0 ➜ False (OFF) 1 ➜ True (ON)			

Logic program editor

Logic functions

6.4.4 Scenario numerico (Numerical scenario)

Description:	When a pulse is received on the TRG input, it runs a sequence of numerical commands, each of which can be set, possibly spacing each command with a preset time, common to all the outputs

Preview:	

Nodes:	TAG	Description		TYPE	IN	OUT
	TRG	Input trigger	*Possible values:* 0 ➜ OFF 1 ➜ ON	T	•	
	RES	Reset scenario Returns all the scenario outputs to the initial status (default).	*Possible values:* 0 ➜ OFF 1 ➜ ON	M	•	
	STOP	Stop scenario If on, the scenario is stopped. The scenario is resumed when the STOP signal is disabled (useful above all when setting time > 0 as an interval between the activation of the scenario outputs).	*Possible values:* 0 ➜ OFF 1 ➜ ON	M	•	
	OUT1 … OUT10	Output 1… 10	*Possible values:* any numerical value	S		•
	TRG1 … TRG10	Trigger 1 … 10	*Possible values:* 0 ➜ OFF 1 ➜ ON	T		•
	+	Add node + trigger				•
Options:	Intervallo uscite	Waiting time between output command	*Possible values:* 1 … 60 (seconds)			
	Set uscita 1 … 10	Value to be set on output 1 … 10	*For outputs from 1 to 10 it is possible to set any value.*			

6.5 Gates

6.5.1 Selettore binario (Binary selector)

Description:	Returns the value of one of the inputs according to the value of the SEL input, which acts as a selector. If SEL=False ➜ OUT=IN0 If SEL=True ➜ OUT=IN1

Preview:	

Nodes:	TAG	Description		TYPE	IN	OUT
	SEL	Input selector	*Possible values:* 0 ➜ OFF (in this case OUT = IN0) 1 ➜ ON (in this case OUT = IN1)	S	•	
	IN0 IN1	Input 0, input 1	*Possible values:* 0 ➜ OFF 1 ➜ ON	M	•	
	OUT	Output	*Possible values:* 0 ➜ OFF 1 ➜ ON	M		•

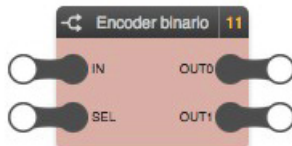*Note: This block performs a similar function to that of a "binary" decoder.*
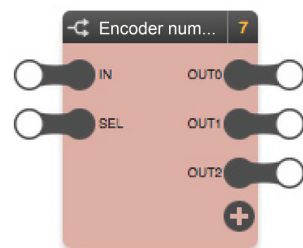
Logic functions

6.5.2 Selettore numerico (Numerical selector)

Description:	Returns the value of one of the inputs according to the value of the SEL input which acts as a selector.
	If SEL=False → OUT=IN0
	If SEL=True → OUT=IN1
Preview:	

Nodes:	TAG	Description		TYPE	IN	OUT
	SEL	Input selector	*Possible values:* 0 → OFF (in this case OUT = IN0) 1 → ON (in this case OUT = IN1)	S	•	
	IN0 IN1	Input 0, input 1	*Possible values:* any numerical value	M	•	
	OUT	Output	*Possible values:* any numerical value	M		•

6.5.3 Encoder binario (Binary encoder)

Description:	Sets one of the two outputs to the input value IN according to the value of the input SEL, which acts as a selector
Preview:	

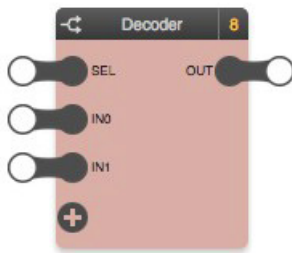| Nodes:	TAG	Description		TYPE	IN	OUT
	IN	Input	*Possible values:* 0 → OFF 1 → ON	S	•	
	SEL	Output selector	*Possible values:* 0 → OFF 1 → ON	S	•	
	OUT0 OUT1	Output 1 (if selector 0) Output 2 (if selector 1)	*Possible values:* 0 → OFF 1 → ON	S		•

6.5.4 Encoder numerico (Numerical encoder)

Description:	Sets the value of IN at one of its outputs according to the value of the SEL input which acts as a selector	*Number of outputs:* from 2 to 10
Preview:		

Nodes:	TAG	Description		TYPE	IN	OUT
	IN	Input	*Possible values:* 0 → OFF 1 → ON	S	•	
	SEL	Output selector	*Possible values:* 1 … 10	S	•	
	OUT0 … OUT9	Output 0… 9	*Possible values:* 0 → OFF 1 → ON	S		•
	+	Add output				•

Example: OUT nodes such as enable can be used for logic networks according to the SEL value.

Logic functions

6.5.5 Decoder

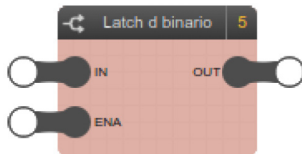| Description:	Restores in output the value of one of the inputs according to the input value of the SEL which acts as a selector				
Preview:					

Nodes:	TAG	Description		TYPE	IN	OUT
	SEL	Input selector	*Possible values:* 0 … 9	S	•	
	IN0 … IN9	Input 0 … 9	*Possible values:* 0 ➜ OFF 1 ➜ ON	S	•	
	OUT	Output	*Possible values:* 0 ➜ OFF 1 ➜ ON	S		•
	+	Add input			•	

Example:

The OUT node can be used to control, or not, an actuator using IN nodes, according to the SEL value
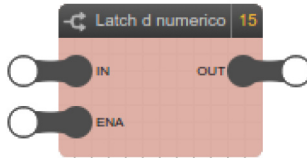
Note: The binary decoder function, i.e. a decoder with 2 binary inputs and binary selection node, is accomplished by the "Binary Switch" logic block.

6.5.6 Latch D binario (Binary D latch)
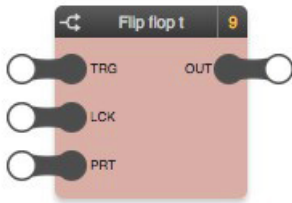
Description:	In this block the input signal IN is propagated to the output OUT if the enable ENA signal is enabled (1). If the enable signal ENA is disabled, the output OUT holds the last state. When enable ENA is again active then (transition 0--> 1) the last value read in the input node IN is sent to the output OUT. Basically, with ENA=0, the Latch block saves the last value read to send it to the output at the moment when ENA is again enabled. The IN and OUT data format is binary.				
Preview:					

Nodes:	TAG	Description		TYPE	IN	OUT
	IN	Input	*Possible values:* 0 … 1	M	•	
	ENA	Enable	*Possible values:* 0 … 1	M	•	
	OUT	Output	*Possible values:* 0 … 1	M		•

Logic functions

6.5.7 Latch D numerico (Numerical D Latch)

Description:	In this block the input signal IN is propagated to the output OUT if the enable ENA signal is enabled (1). If the enable signal ENA is disabled, the output OUT holds the last state. When enable ENA is again active then (transition 0--> 1) the last value read in the input node IN is sent to the output OUT. Basically, with ENA=0, the Latch block saves the last value read to send it to the output at the moment when ENA is again enabled. The IN and OUT data format is numerical.
Preview:	

Nodes:	TAG	Description			TYPE	IN	OUT
	IN	Input	*Possible values:*	any numerical value	M	•	
	ENA	Enable	*Possible values:*	0 … 1	M	•	
	OUT	Output	*Possible values:*	any numerical value	M		•

6.5.8 Flip-flop T (T flip-flop)

Description:	T flip-flop
	Works as a step-step relay. Whenever a rising edge is proposed to the input (TRG) the output (OUT) changes status. If the LCK input (Block) is 1 (True) the effect of the TRG is inhibited and the output never changes. If the input PRT (Priority) is 1 the output takes on the value set in the parameter "Priority value".
	It may be used for example to control the lights in a corridor. The light can be controlled normally only if a brightness threshold is satisfied (this condition would end up in the LCK) and if it is in any case always on at night (Flag connected to the PRT input)
Preview:	

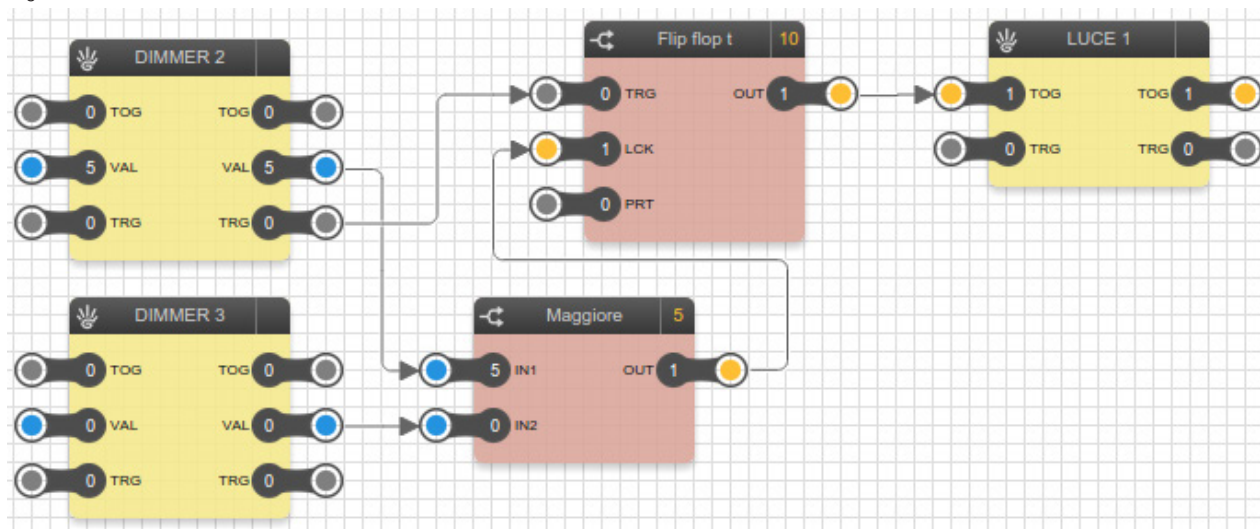| Nodes:	TAG	Description			TYPE	IN	OUT
	TRG	Trigger	*Possible values:*	0 ➜ OFF 1 ➜ ON	T	•	
	LCK	Locks the current state	*Possible values:*	0 ➜ OFF 1 ➜ ON	S	•	
	PRT	Priority flag	*Possible values:*	0 ➜ OFF 1 ➜ ON	S	•	
	OUT	Output signal	*Possible values:*	0 ➜ OFF 1 ➜ ON	M		•
Options:	Valore priorità	Value to assign to the output for a priority flag		*Possible values:* VERO/FALSO (TRUE/FALSE)			

Truth table:

	TRG	OUT
With LCK=0	0>1	NOT OUT
With LCK=1	0>1	Does not change

Note: See also the Priority flag parameter.

Logic functions

Example:

A light is controlled via a T Flip Flop, which measures the change in state of a dimmer and stops it from being switched on if the value of the first dimmer is greater than the second.


6.5.9 Flip-flop RS (RS Flip-flop)

Description:	RS flip-flop
	An elementary memory block which is "loaded" with the SET input and reset with the RES input (reset). If both inputs are 1 the one specified by the parameter "Selection priority" will prevail.
	It may for example be used to manage an alarm signal. An alarm contact is connected to the SET. When it has gone to 1 the Flip-Flop maintains the output on 1 until it has been reset by the RES. In this way even if the alarm is "normalised" (moving to 0) the information is maintained.
Preview:	

Nodes:	TAG	Description		TYPE	IN	OUT
	SET	Set	*Possible values:* 0 ➜ OFF 1 ➜ ON	M	•	
	RES	Reset	*Possible values:* 0 ➜ OFF 1 ➜ ON	M	•	
	OUT	Output signal	*Possible values:* 0 ➜ OFF 1 ➜ ON	M		•
Options:	Priorità selezione (selection priority)	*Possible values:* Set / Reset				

Truth table:

S	R	OUT
0	0	Does not change
0	1	0
1	0	1
1	1	See selection priority parameter

Example:

A light is switched on if there is an alarm signal. The light connected to the RES input is used to reset the Flip Flop RS status.

Logic functions
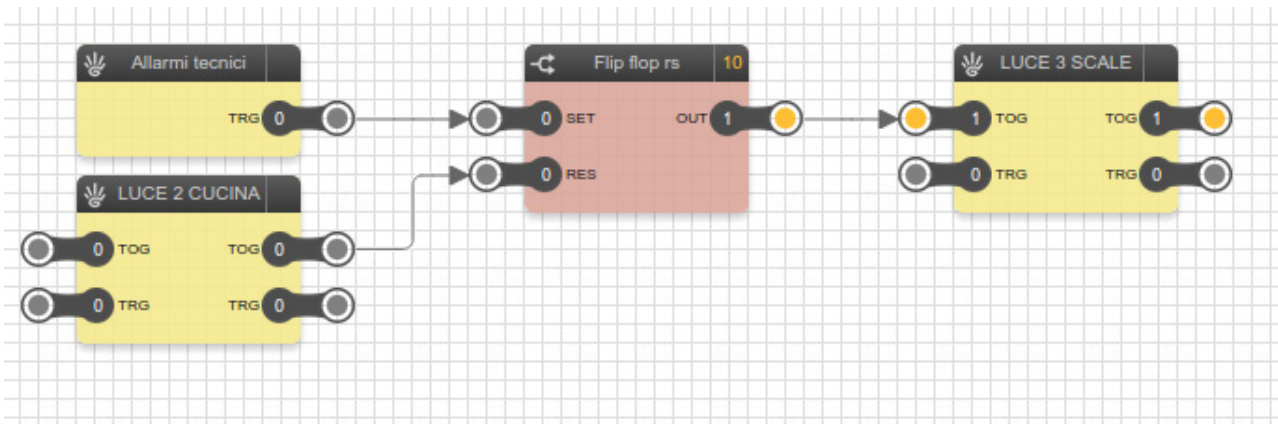
6.5.10 Flip-flop D (D flip-flop)

Description:	D-type flip-flop
	Operation is similar to that of a D Latch, with the difference that the D Flip-Flop operates on a change of the CLK edge. The data DAT is transferred to OUT only on the rising edge of the CLK signal and holds it until the next rising edge of CLK (this block is basically a memory cell).
Preview:	

Nodes:	TAG	Description		TYPE	IN	OUT
	DAT	Data	*Possible values:* 0 … 1	M	•	
	CLK	Clock	*Possible values:* 0 … 1	T	•	
	OUT	Output	*Possible values:* 0 … 1	M		•

6.6 Comparisons

6.6.1 Comparison operators

Description:	*Available operators:* • Maggiore (Greater than) • Maggiore uguale (Greater than or equal to) to) Compares the value of the two inputs, and outputs a TRUE/FALSE • Minore (Less than) value, depending on the specific operator • Minore uguale (Less than or equal to) • Uguale (Equal to) • Diverso (Different)
Preview:	

Nodes:	TAG	Description		TYPE	IN	OUT
	IN1 IN2	Input 1, input 2	*Possible values:* any numerical value	S	•	
	OUT	Result of comparison	*Possible values:* 0 ➜ OFF 1 ➜ ON	S		•

Logic program editor

VIMAR

Logic functions
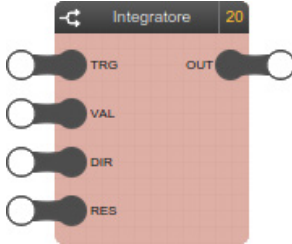
6.7 Operations

6.7.1 Mathematical operators

Description:	Performs a mathematical operation on the inputs, depending on the operator type	*Available operators:*	• Massimo (Maximum) • Minimo (Minimum) • Media (Average) • Somma (Addition) • Sottrazione (Subtraction) • Moltiplicazione (Multiplication) • Divisione (Division) • Valore assoluto (Absolute value) • Integratore (integrator) • Range • Log10

Preview:	

Nodes:	TAG	Description		TYPE	IN	OUT
	IN1 IN2 (*) ...	Input 1, input 2 ...	*Possible values:* any numerical value	S	•	
	OUT	Value (result of the operation)	*Possible values:* any numerical value	S		•
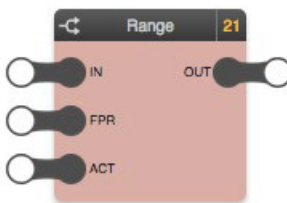
(*) The number of outputs may be limited depending on the operation (e.g., division max 2, absolute value max 1)

6.7.2 Integratore (Integrator)
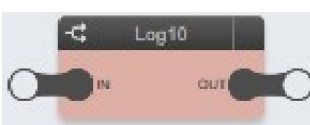
Description:	A logic block which performs an integration function on a numerical value. For each edge on the TRG node it adds the content of the input VAL node (positive or negative). The output numerical node VAL carries the integrated value. A DIR node will also be added that adds or subtracts the input value VAL (to be considered as a value with a sign)

Preview:	

Nodes:	TAG	Description			TYPE	IN	OUT
	TRG	Trigger	*Possible values:*	0 ➜ OFF 1 ➜ ON	T	•	
	VAL	Value	*Possible values:*	any numerical value	S	•	
	DIR	Counter direction	*Possible values:*	0 ➜ ADDITION 1 ➜ SUBTRACTION	S	•	
	RES	Reset ter)	*Possible values:*	0 ➜ OFF 1 ➜ ON (resets the coun-	M	•	
	OUT	Value (result of the operation)	*Possible values:*	any numerical value	S		•

Logic program editor

Logic functions

6.7.3 Ranges

Description:	Performs linear interpolation of the input value IN according to the assigned mapping, also known as "characteristic", defined by two pairs of values (X,Y). The value IN is a ratio of between X0 and X1, and this ratio is in turn calculated between Y0 and Y1 to determine the output value.
	If priority mode is set, a predetermined value is returned.
	The typical field of application of this block is the conversion of values between different variables.
Preview:	

Nodes:	TAG	Description		TYPE	IN	OUT
	IN	Input	*Possible values:* any numerical value	S	•	
	FPR	Priority enable	*Possible values:* 0 ➜ No priority 1 ➜ Priority (the priority value is returned)	S	•	
	ACT	Direct/reverse operation	*Possible values:* 0 ➜ Direct operation 1 ➜ Inverse operating	S	•	
	OUT	Output	*Possible values:* any numerical value	S		•
Options:	X0 Y0 X1 Y1	Characteristics of the linear interpolation	*Possible values:* any numerical value			
	Priority value	Value to be returned in case of priority enabling				

6.7.4 Log10

Description:	A logic block that performs a base 10 logarithm on a numerical value.
Preview:	

Nodes:	TAG	Description		TYPE	IN	OUT
	IN	Input	*Possible values:* any numerical value	S	•	
	OUT	Output	*Possible values:* any numerical value	S		•

Logic program editor
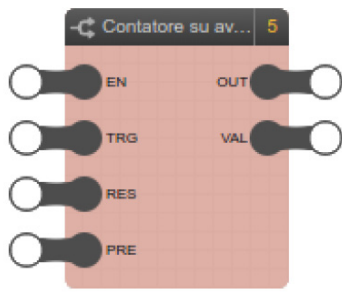
Logic functions

6.8 Counters

6.8.1 Contatore SU, contatore GIÙ (UP counter, DOWN counter)
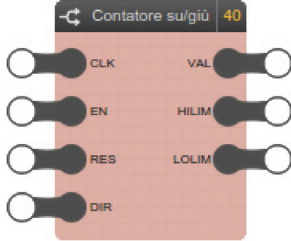
Description:	Counts the number of pulses received at the input (trigger), increasing or decreasing its value each time (depending on the type of counter). *Types of counter:* up counter, down counter.
Preview:	

Nodes:	TAG	Description		TYPE	IN	OUT
	EN	Enable	*Possible values:* 0 ➜ Not enabled 1 ➜ Enabled	S	•	
	TRG	Trigger	*Possible values:* 0 ➜ OFF 1 ➜ ON (the counter is incremented)	T	•	
	RES	Reset	*Possible values:* 0 ➜ OFF 1 ➜ ON (resets the counter)	M	•	
	OUT	Output	*Possible values:* 0 ➜ OFF 1 ➜ ON	M		•
	VAL	Current value	*Possible values:* any numerical value	S		•

Options:		
	Preset	Default value, set when resetting or starting the logic element. In the case of a counter UP the count starts from 0 and must reach Preset to enable OUT, while in Counter DOWN the count starts from Preset and must reach 0 to enable OUT. *Possible values:* 32767

6.8.2 Contatore SU avanzato/GIÙ avanzato (Advanced UP/Advanced DOWN Counter)
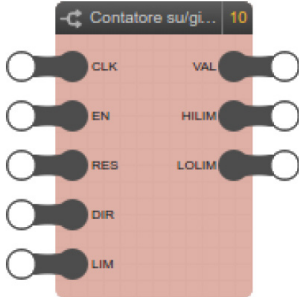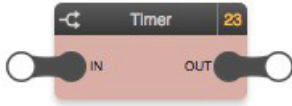
Description:	The function is similar to that of the UP Counter, DOWN Counter with the difference that some options have been converted into nodes so that they can be changed dynamically via the virtual datapoints
Preview:	

Nodes:	TAG	Description		TYPE	IN	OUT
	EN	Enable	*Possible values:* 0 ➜ OFF 1 ➜ ON	S	•	
	TRG	Trigger	*Possible values:* 0 ➜ OFF 1 ➜ ON	T	•	
	RES	Reset	*Possible values:* 0 ➜ OFF 1 ➜ ON (resets the counter)	M	•	
	PRE	Preset	*Possible values:* 0 ... 32767	S	•	
	OUT	Output	*Possible values:* 0 ➜ OFF 1 ➜ ON	M		•
	VAL	Current value	*Possible values:* any numerical value	S		•

6.8.3 Contatore SU/GIÙ (UP/DOWN counter)

Description:	Counts the number of pulses received at the input (trigger), increasing or decreasing its value each time (depending on the type of counter). *Types of counter:* up counter, down counter.
Preview:	

Nodes:	TAG	Description		TYPE	IN	OUT
	CLK	Clock	*Possible values:* 0 ➜ OFF 1 ➜ ON (the counter is incremented)	T	•	
	EN	Enable	*Possible values:* 0 ➜ Not enabled 1 ➜ Enabled	S	•	
	RES	Reset	*Possible values:* 0 ➜ OFF 1 ➜ ON (resets the counter)	M	•	
	DIR	Counter direction	*Possible values:* 0 ➜ UP 1 ➜ DOWN	S	•	
	VAL	Current value	*Possible values:* any numerical value	S		•
	HILIM	High limit reached. Trigger signal		T		•
	LOLIM	Low limit reached. Trigger signal		T		•
Options:	Limite massimo	Default value, set when resetting or starting the logic element. In the case of an UP counter the count starts from 0 and must reach Maximum Limit to enable OUT, while in a DOWN counter the count starts from Maximum Limit and must reach 0 to enable OUT. *Possible values:* 32767				

6.8.4 Contatore SU/GIÙ avanzato (Advanced UP/DOWN counter)

Description:	The function is similar to that of the UP/DOWN Counter with the difference that some options have been converted into nodes so that they can be changed dynamically via the virtual datapoints
Preview:	

Nodes:	TAG	Description		TYPE	IN	OUT
	CLK	Clock	*Possible values:* 0 ➜ OFF 1 ➜ ON	T	•	
	EN	Enable	*Possible values:* 0 ➜ OFF 1 ➜ ON	S	•	
	RES	Reset	*Possible values:* 0 ➜ OFF 1 ➜ ON (resets the counter)	M	•	
	DIR	Counter direction	*Possible values:* 0 ➜ UP 1 ➜ DOWN	S	•	
	LIM	Maximum limit	*Possible values:* any numerical value	S	•	
	VAL	Current value	*Possible values:* any numerical value	S		•
	HILIM	High limit reached	*Possible values:* 0 ➜ OFF 1 ➜ ON	T		•
	LOLIM	Low limit reached	*Possible values:* 0 ➜ OFF 1 ➜ ON	T		•

Logic program editor

Logic functions

6.9 Timers and scheduling

6.9.1 Timer

Description:	Delays the value received at the input by a preset time
	When a 1 is received in the input IN (rising edge), an internal counter starts until the time specified as "rising delay", after which the output is moved to 1; vice versa, on receipt of an input 0 (falling edge), the block waits for the time set as "falling delay" before setting the output to 0.

Preview:

Nodes:	TAG	Description		TYPE	IN	OUT
	IN	Input ON/OFF signal	*Possible values:* 0 → OFF 1 → ON	M	•	
	OUT	Output ON/OFF signal, delayed by the timer	*Possible values:* 0 → OFF 1 → ON	M		•
Options:	Ritardo in salita	Delay in propagating the rising edge received at the input	*Possible values:* from 1 second to 12 hours in steps of 1 second.			
	Falling delay	Delay in propagating the falling edge received in input	*Possible values:* from 1 second to 12 hours in steps of 1 second.			

6.9.2 Timer avanzato (Advanced timer)

Description:	The function is similar to that of the Timer with the difference that some options have been converted into nodes so that they can be changed dynamically via the virtual datapoints

Preview:

Nodes:	TAG	Description		TYPE	IN	OUT
	IN	Start sequence	*Possible values:* 0 → OFF 1 → ON	M	•	
	D.UP	Up delay	*Possible values:* any numerical value	S	•	
	D.DWN	Down delay	*Possible values:* any numerical value	S	•	
	OUT	Output delayed by timer	*Possible values:* 0 → OFF 1 → ON	M		•

CAUTION:
The value provided to the Timer Avanzato logic block to define the up or down delay to be applied (via the D.UP and D.DWN nodes) must always be expressed in seconds.

Logic program editor

Logic functions

6.9.3 Crono settimanale (Weekly timer)

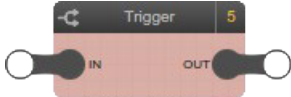| Description:	Used to set weekly scheduling				
	The block adopts a value of 1 or 0 depending on the time and day of the week, according to the programming set in the editor or by the end user.				
Preview:					
Nodes:	**TAG**	**Description**	**TYPE**	**IN**	**OUT**
	OUT	Scheduling status *Possible values:* 0 ➔ OFF 1 ➔ ON	S		•
Options:	Pianifica	Scheduling button, used to open the scheduling pop-up to set when the output has to be set to ON			

Clicking the "PIANIFICA" button opens the pop-up used to establish the times at which the lock should be ON for each day of the week, in steps of 10 minutes:

Daily profile preview

Status change buttons for current interval

Change day of the week

Duplicate current profile

Moving time interval selector

Copy current interval to the next one

Logic functions

6.9.4 Crono periodico (Periodic timer)

Description:	Used to set periodic scheduling, consisting of one or two intervals for each day of the week				
	The block adopts a value of 1 or 0 depending on the time and day of the week, according to the programming set in the editor or by the end user.				
Preview:					
Nodes:	TAG	Description	TYPE	IN	OUT
	OUT	Scheduling status　　　　　　　*Possible values:*　0 ➔ OFF　1 ➔ ON	S		•
Options:	Pianifica	Scheduling button, used to open the scheduling pop-up to set when the output has to be set to ON			

Clicking the "PIANIFICA" button opens the pop-up used to establish one or two intervals for each day of the week during which the scheduling is active:

Select day of the week

Start and end time of the first event

Start and end time of the second event
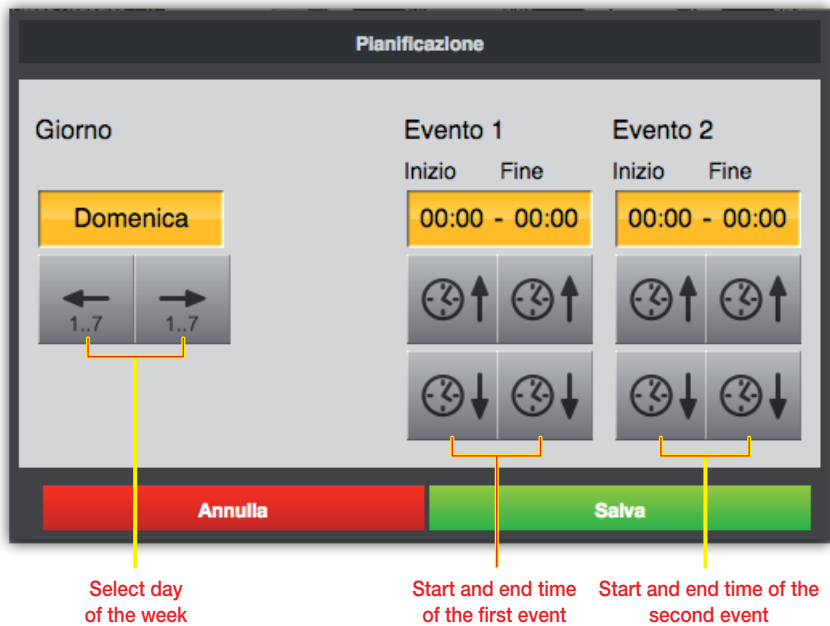
In addition to the increase and decrease buttons, you can also change the times by clicking on the time indicators: a pop-up prompts you to enter the start or end time of the event.

Logic program editor

Logic functions

6.9.5 Crono ciclico (Cyclic timer)

Description:	Used to set cyclic scheduling, based on an ON time and an OFF time				
	The block adopts a value of 1 or 0 depending on the duration of the cycle set in the editor or by the end user.				
Preview:					

Nodes:	TAG	Description		TYPE	IN	OUT
	OUT	Scheduling status	*Possible values:* 0 → OFF 1 → ON	S		•
Options:	Pianifica	Scheduling button; used to open the scheduling pop-up to set the ON and OFF times				

Clicking the "PIANIFICA" button opens the pop-up used to establish the ON and OFF time:

Setting ON time   Setting OFF time
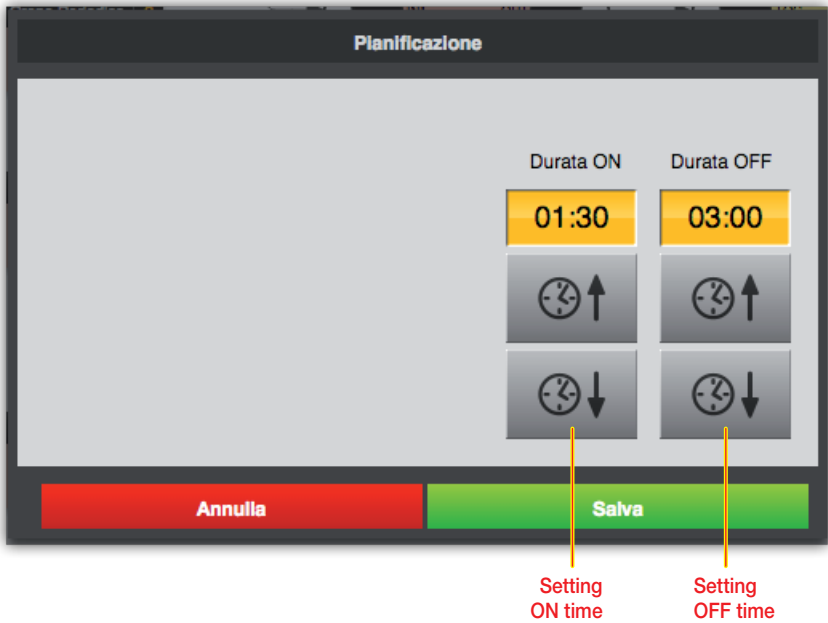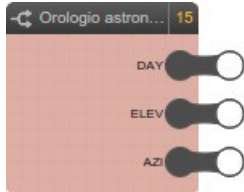
As for periodic programs, clicking the time indicators will open the pop-up used to enter the ON/OFF time as an alternative to increasing/decreasing it with the buttons.

Logic functions

6.9.6 Trigger

Description:	Generates a trigger (pulse of the duration of one cycle) on an edge detected at the input
	When it receives a 1 on input or a 0 (depending on the value set in the parameter "Edge"), the output set to 1 for the duration of a single processing cycle, then the output is once again set to 0. In this way, it is possible to generate an "impulse" for logic blocks which require one (e.g. scenarios, sequencer etc.) on the input rising edge.

Preview:	

Nodes:	TAG	Description	TYPE	IN	OUT
	IN	Input edge	M	•	
	OUT	Pulse of the duration of one cycle. The logic is run repeatedly over time, the pulse generated by the trigger lasts only for one run cycle, on the next step if a new edge is not measured on input nopulse will be generated.	T		•
Options:	Fronte	Rising or falling edge to be detected at the input			

6.9.7 Orologio astronomico (Astronomical clock)

Description:	Returns the status of day/night, the degree of elevation of the sun and the angle to the North according to the current date/time and the coordinates given via parameters.

Preview:	

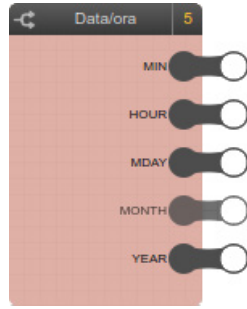| Nodes:	TAG	Description		TYPE	IN	OUT
	DAY	Day	Possible values: 0 ➜ Night 1 ➜ Day	S		•
	ELEV	Elevation of the sun Possible values: -90° … +90° Returns the height of the sun over the horizon. The value 0° indicates the horizon. Positive values indicate day, negative ones night.		S		•
	AZI	Azimuth Possible values: 0° … 360° Returns the angular distance of the sun from the North. The value 0° indicates the North, 90° East, 180° South and 270° West.		S		•
Options:	Latitudine	Latitude (-90/90) with up to 7 decimal digits Latitude coordinate example: Rome 41.9100711				
	Longitudine	Longitude (-180/180) with up to 7 decimal digits Longitude coordinate example: Rome 12.5359979				
	Soglia	Threshold related to the sun's elevation to determine the day/night output (default 0°, in this case when the sun passes the horizon the DAY output will be day)				
	Fuso orario (Time zone)	Can be selected in the drop down menu				
	Cambio ora legale (Daylight saving time change)	Possible values: disabilitato (disabled), automatico (automatic), and manuale (manual). In manual mode, the time/daylight saving time can be set.				

CAUTION: This logic block needs the system clock in the home automation system.
If the system clock is configured to automatically send the daylight saving time, it is necessary to set the "Cambio ora legale" (Change daylight saving time) parameter in the device.
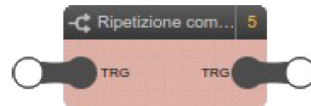
It is possible to simulate the Orologio Astronomico (Astronomical clock) block by changing the date/time settings in the advanced options menu. For a correct simulation, the "Cambio ora legale" parameter must be set to automatic; at the end of the simulation, then restore this parameter to disabled (as described in the previous note).
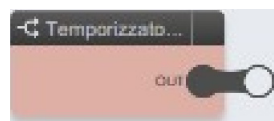
6.9.8 Data/Ora (Date/Time)

Description:	Returns the current time of the By-me system.				
Preview:					
Nodes:	**TAG**	**Description**	**TYPE**	**IN**	**OUT**
	MIN	Minutes *Possible values:* 0…59	S		•
	HOUR	Hours *Possible values:* 0…23	S		•
	MDAY	Day of the month *Possible values:* 1…31	S		•
	MONTH	Month *Possible values:* 1…12	S		•
	YEAR	Year *Possible values:* 2015…2099	S		•

6.9.9 Ripetizione comando (Command repeat)

Description:	It features two trigger nodes; when a trigger reaches the TRG input it generates N+1 triggers on the output TRG node that are distanced from each other by a configurable parameter.				
Preview:					
Nodes:	**TAG**	**Description**	**TYPE**	**IN**	**OUT**
	TRG	Trigger for sending/receiving from/to bus	T	•	•
	TRG	Trigger for sending/receiving from/to bus	T	•	•
Options:	Intervallo (s)	Time in seconds between the generation of one output trigger and the next one			
	Ripetizioni	Number of triggers to be generated after the first one			

6.9.10 Temporizzatore astronomico (Astronomical timer)

Description:	Logic block only available in logics created with the home automation system gateway. Not provided in the 01468 logic unit.				
Preview:					
Nodes:	**TAG**	**Description**	**TYPE**	**IN**	**OUT**
	OUT	Output that controls an actuation.	S		•

The parameters of the Temporizzatore Astronomico (Astronomical timer) block are the same as those of the Orologio Astronomico (Astronomical clock); in addition, the "Start" and/or "Stop" modes can be parametrised (as "Alba" (Sunrise) or "Tramonto" (Sunset)) with "Modalità evento" (Event mode), and "Stop" in "Modalità durata" (Duration mode).

CAUTION: This logic block needs the system clock in the home automation system.
We recommend you use the "Time change" parameter in the Orologio Astronomico (Astronomical Clock) logic block in "Automatic" mode.

It is possible to simulate the Temporizzatore Astronomico (Astronomical timer) by changing the date/time settings in the advanced options menu. For a correct simulation, the "Cambio ora legale" parameter must be set to automatic; at the end of the simulation, then restore this parameter to disabled (as described in the previous note).

Logic functions

LAT: Latitude (-90°/90°) with up to 7 decimal digits. Latitude coordinate example: Rome 41.9100711

LONG: Longitude (-180°/180°) with up to 7 decimal digits. Longitude coordinate example: Rome 12.5359979

TZ: Time Zone; time difference with respect to GMT; Possible values from -12 to +13 in steps of 0.5 (e.g. Adelaide = 9.5)

TM: Time Mode; disabled = 0, automatic = 1.

START: start reference event (sunrise or sunset)

START_DIFF: advance (negative) or delay (positive) with respect to the start event (expressed in minutes)

MODE: mode to define the on duration: Duration (an on time is set) or Event (end with reference to a solar event after the start, i.e. the next sunrise or sunset).
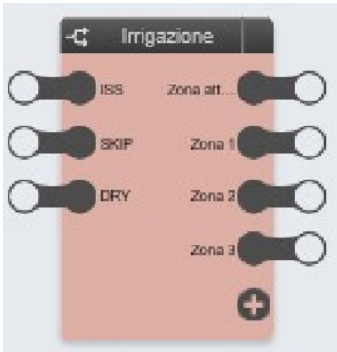
DURATION: on time (if the corresponding operating mode has been chosen)

STOP: stop reference event (sunrise or sunset)

STOP_DIFF: advance (negative) or delay (positive) with respect to the stop event (expressed in minutes)

Logic program editor

Logic functions

6.9.11 Irrigazione (Irrigazione)

Description:	Logic block only available in logics created with the home automation system gateway. Not provided in the 01468 logic unit.				
Preview:					
Nodes:	**TAG**	**Description**	**TYPE**	**IN**	**OUT**
	ISS	Trigger (associated with a key) to start or stop the function independently from the "Calendar" (it bypasses the programmed start time). If the sequence is stopped, pressing the key activates it immediately; if the sequence is running, the key stops it immediately and it then restarts with the next calendar cycle.		•	
	SKIP	Trigger to end the current output activity and go straight to the next output in advance. If the sequence is at the last output, the command has no effect.		•	
	DRY	Input to transmit information used to disable irrigation (e.g. the output from the weather station, and contact interface connected to a ground moisture sensor, etc.); it is therefore external information that affects the normal set operation (e.g. a time schedule), but that does not affect ISS and SKIP commands as they are "manual" functions.		•	
	Zona 1..Zona 16	Zones into which the irrigation system is divided (up to a maximum of 16).			•

Important: The ISS and SKIP inputs are associated with art. 01480, 01481, 01482, 01485, 01486, 01487, 01488 and 01489.

A value of 0 (default) means normal operation (i.e. no conditions that can disable irrigation). An input value of 1 disables the irrigation.

Zone 1: Actuator command to irrigate zone 1

....

Zone 16: Actuator command to irrigate zone 16

Active Zone: This indicates the zone that has been controlled correctly (numeric value 0..16; 0 means no zone has been controlled, i.e. no irrigation in progress)

Clicking the "PIANIFICA" button opens the pop-up used to set one or two intervals for each day of the week during which irrigation is active:

6.10 Variables
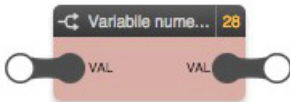
6.10.1 Introduction

As explained in section 3.11, the variables enable passing values between different programs. The variables must first be created with the "+" button in the corresponding section of the main menu, then they can be dragged into the programs that need to use them.

6.10.2 Variabili binarie (Binary variables)

Description:	Used to transfer a boolean value between different programs.				
Category:	Binary variables				
Preview:					
Nodes:	TAG	Description		TYPE	IN	OUT
	VAL	Value to assign to the variable	*Possible values:* 0 → OFF 1 → ON	M	•	
	VAL	Current value of the variable	*Possible values:* 0 → OFF 1 → ON	M		•

6.10.3 Variabili numeriche (Numerical variables)

Description:	Used to transfer a boolean value between different programs.				
Category:	Binary variables				
Preview:					
Nodes:	TAG	Description		TYPE	IN	OUT
	VAL	Value to assign to the variable	*Possible values:* any numerical value	M	•	
	VAL	Current value of the variable	*Possible values:* any numerical value	M		•

7. Simulation

7.1 Introduction

After creating a logic program, you can simulate its operation in the editor, manually entering the state of the inputs and verifying the output processing in real time, also with the logic blocks that entail changing the outputs over time.

7.2 Types of simulation

Two different types of simulation are available:

- **Continuous simulation**: the programs run in the background, and are affected by changes of state of the nodes in real time.
- **Step-by-step simulation**: each program cycle must be launched manually and it is possible to change the state of the nodes between cycles.

The first type provides a more realistic assessment of the logic networks created, the second one permits a thorough check on every single passage of values between blocks and offers a higher level of diagnosis.

7.3 Graphical simulation environment

On pressing one of the simulation buttons (continuous or step-by-step), the editor window is modified as follows:

- The main menu is limited to PROJECT view only, allowing only passing between logic programs. It is not possible to create or delete programs.
- The "Proprietà generali" pane is closed to provide the most workspace for the simulation.
- Each drag&drop operation, connection, change or deletion of the content of the logic programs is locked.
- The nodes are coloured according to their state and allow forcing the value manually (as detailed below).

The colour of the nodes follows this convention:

Binary nodes	Grey	Value 0 (OFF)
	Yellow	Value 1 (ON)
Numerical nodes	Blue	Any value of those permitted

During simulation, the editor shows a variety of information in the message area related to the running of the programs, the manual state changes (made by the user) and automatic state changes (detected by the logic blocks). In addition, during step-by-step simulation, many "debug" level messages are reported, allowing a thorough analysis of the programs running, especially useful in the event of errors or malfunctions.

The message area, normally closed to provide the most space for the simulation, can be opened to view these messages, the number of which – depending on the type – is summarized on the right of the message bar, visible even when closed. For more details on the message area, see section 2.7.

The following figure shows an example of simulation with the message area open:

VIMAR
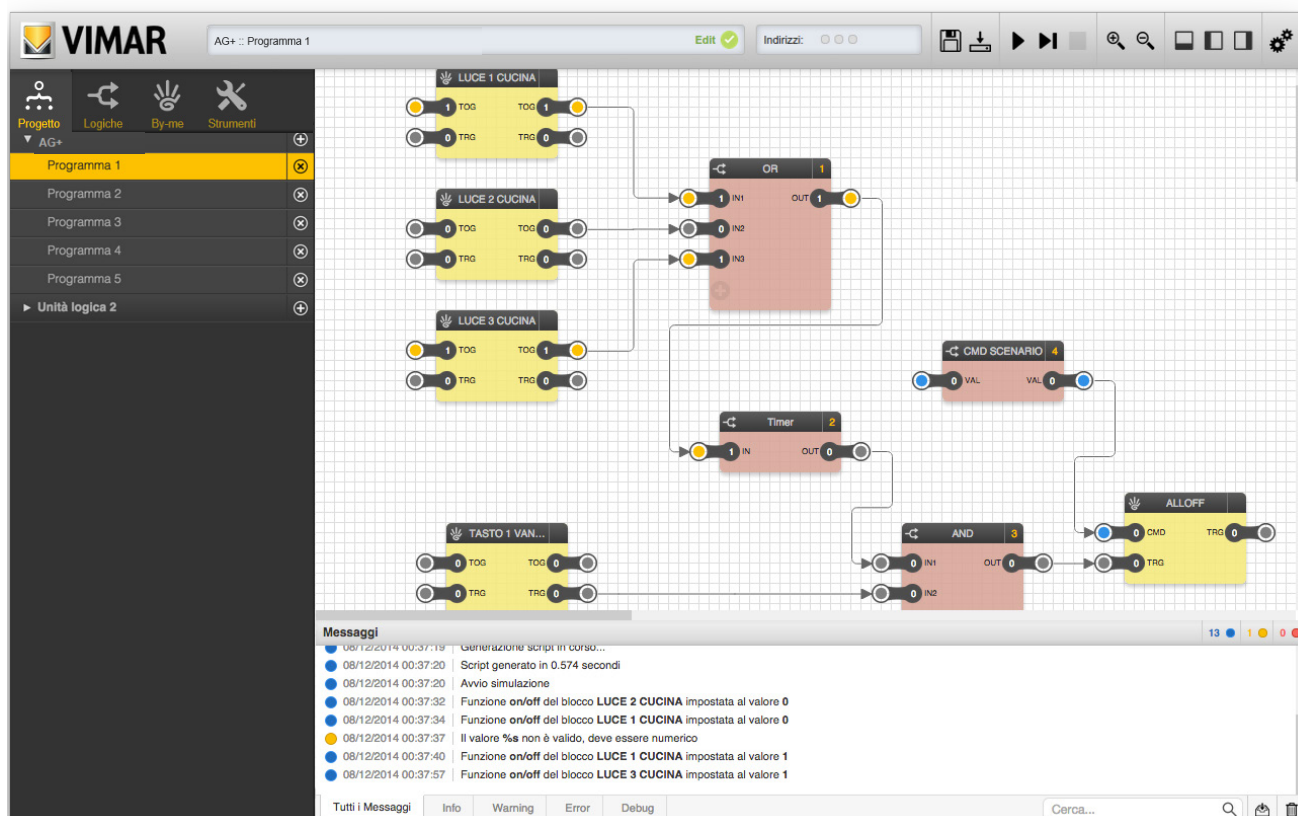
Simulation

7.4 Manual data input

To set the status of a node manually, do the following:

- Double click on the value of the node (the label becomes editable)
- Delete the current value and enter the new value
- Press ENTER

The colour of the node (if digital) changes as a function of the new value, and is passed to the simulator, which propagates it instantly (in the case of continuous simulation) or at the next cycle (in step-by-step mode).

The output values from the blocks may be modified but not the inputs.

The inputs of a non-connected block (e.g. the input of a "Greater than" comparison block used as a threshold) cannot be modified during simulation.

During simulation they maintain the default value set during logic editing.

7.5 Simulating sending a signal from a trigger node

In both simulation modes it is possible to generate a rising edge from a trigger node by double clicking on the node itself. As the trigger signal stays on 1 only for the duration of the run cycle, the visual feedback, above all in continuous simulation, may be very short.

7.6 Stopping the simulation

You can stop the simulation at any time by pressing the simulation stop button on the toolbar (normally not accessible outside simulation).

Drawing tools

8. Drawing tools
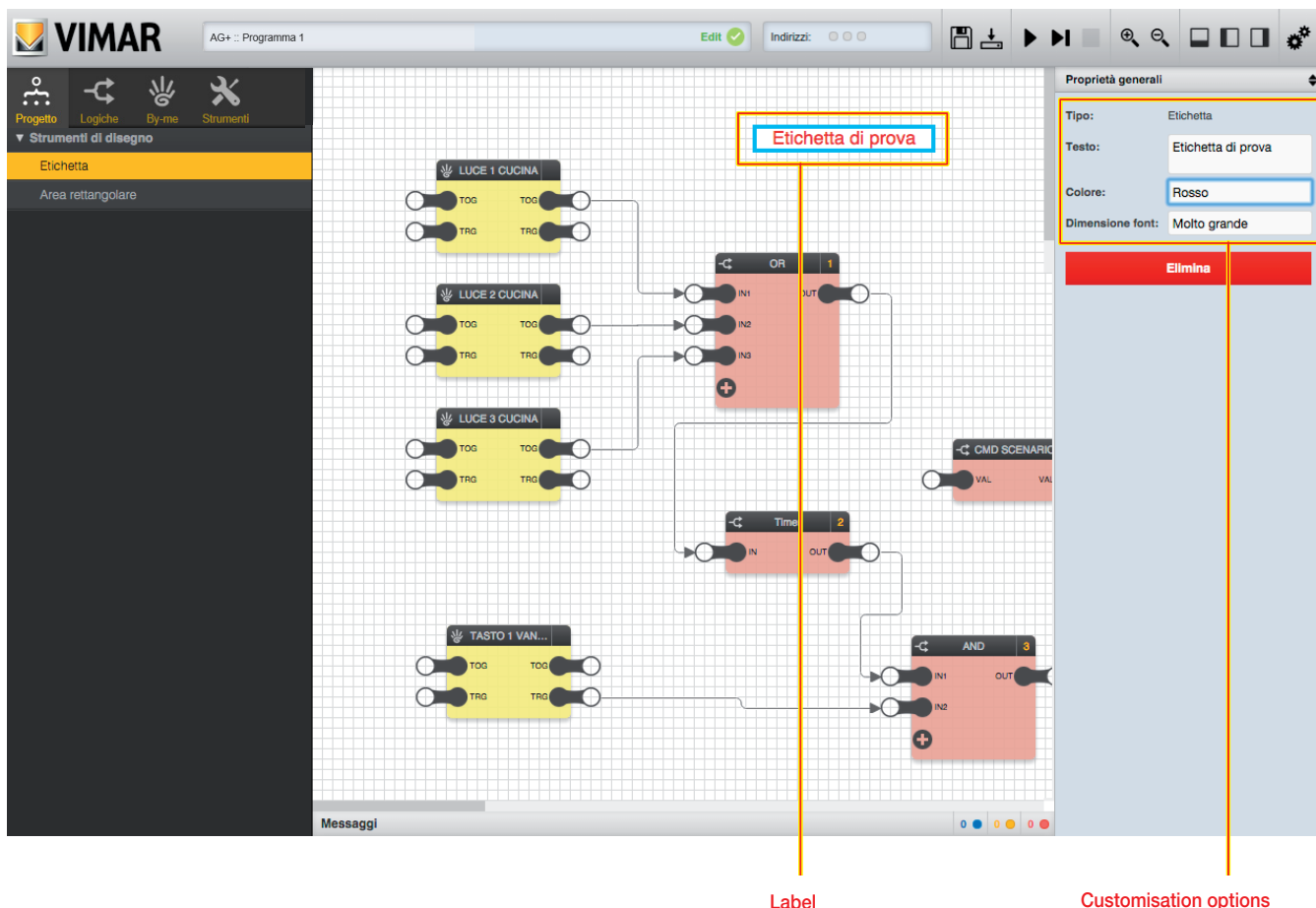
8.1 Introduction

In order to increase the readability of the logic programs, especially with complex logic networks, the editor provides several drawing tools with which the user can add notes and highlight areas of the program.

These tools are available in the "STRUMENTI" area of the main menu in the "STRUMENTI DI DISEGNO" section; they can be dragged into the logic programs using *drag&drop* in the same way as for other types of objects, as seen previously.

8.2 Etichetta (Label)

Labels let you enter free text in programs; you can enter an unlimited number of labels for each logic program.

Label   Customisation options

After dragging a label into the logic program and placing it at the desired point, it can be customised by opening the "Proprietà generali" pane (after selecting it); the following options are available:

Testo	Text displayed in the logic program
Colore	Lets you choose the colour of the text
Dimensione font	Lets you choose the font size

Labels can be removed from logic programs by selecting them and using the corresponding "ELIMINA" button in the "Proprietà generali" pane, or pressing the DEL key on the keyboard.

Logic program editor

VIMAR

Drawing tools

8.3 Area rettangolare (Rectangular area)

You can highlight one or more portions of the logic program by dragging from the main menu the same number of coloured rectangular areas, as exemplified in the following figure:
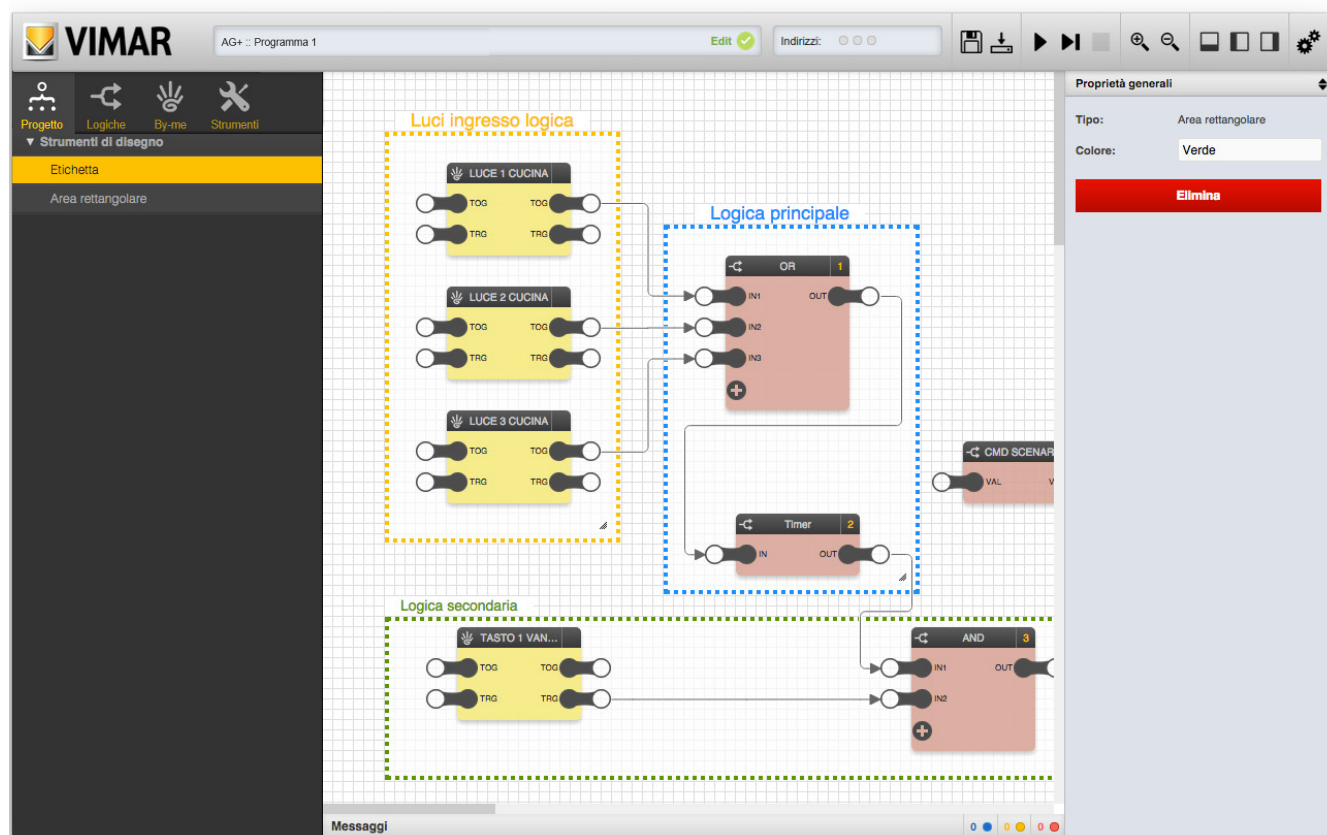
After dragging a rectangular area into a program, you can:

- Resize it by dragging the slider at the bottom right corner.
- Change its border colour using the "colore" selector in the "Proprietà generali" pane.

The rectangular areas are always drawn underneath the blocks and their connections; moreover, they do not support multiple selection (like blocks or labels), so to customise them or remove them from the program it is necessary to click them one at a time and use the tools in the "Proprietà generali" pane ("colore" selector and "ELIMINA" button to remove them from the program).

9. Application examples

This section illustrates, as an example, the construction of some logic programs to implement typical functions of the By-me system.

9.1 Activating a scenario following an intrusion alarm

If there is an evening or night-time alarm (between 18:00 and 8:00 am the following day), the logic program enables a lights on SCENARIO which is then switched off 1 hour later via a lights off scenario.
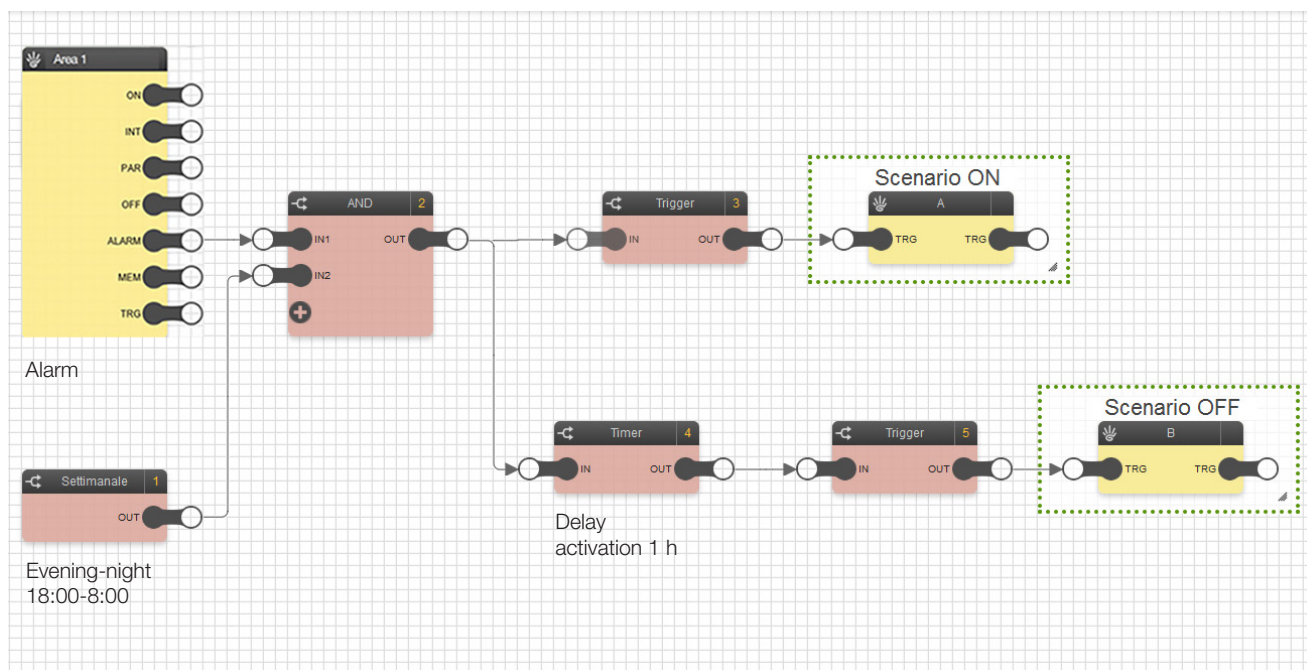
- The Area 1 block of the By-alarm system and the CRONO SETTIMANALE block are used, and must be connected to the respective inputs of the AND logic; a scheduling is set in the CRONO SETTIMANALE in which the block is ON between 18:00 and 23.59 of the current day, and between 0:00 and 8:00 of the next day.
- When an intrusion alarm is triggered during the time set in the CRONO SETTIMANALE, the OUT output of the AND logic switches on the SCENARIO ON block (lights on) and the TIMER block, which enables the SCENARIO OFF (lights off) after 1 h (value set as the rising delay).
- The SCENARIO blocks are always preceded by TRIGGER blocks with the FRONTE parameter set to TRUE.

Logic program editor

Application examples

9.2 Sequential and timed irrigation with start/lock control from button

The logic program switches on the irrigation in 6 separate zones of the garden, switching them on in sequence for 10 minutes in each zone.
In addition to automatic switch-on, the irrigation can also be controlled manually via a button; this button can also be used to block the irrigazione prior to the natural end of the sequence.
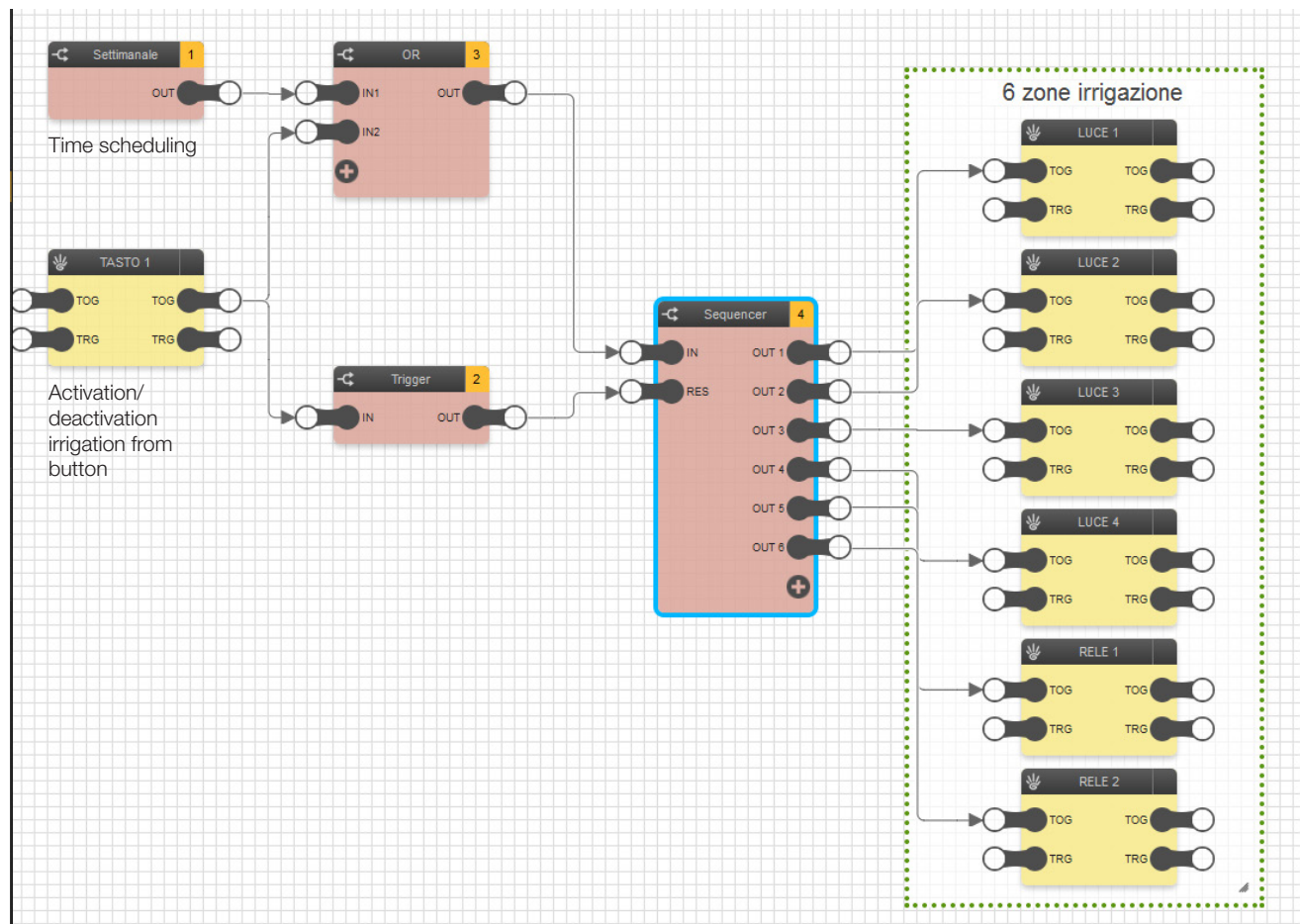
- Use the TASTO block and the CRONO SETTIMANALE block, connected to inputs IN1 and IN2 respectively in the OR logic; the CRONO SETTIMANALE has a scheduling in which the block is ON every day at a set time (for example, 3 pm) for the length of time needed to run the various programs (60 min. in this case) since we want to control 6 zones for 10 minutes each.

- The BUTTON block can switch on the irrigation regardless of the set scheduling.

- The OR logic is connected to the SEQUENCER block input, the outputs of which (OUT1...OUT6) are connected to the respective ON/OFF blocks that enable the irrigation zones.

- The BUTTON block is also connected through the TRIGGER block to the RES input to stop the activation sequence in the 6 zones; the TRIGGER is on with the parameter EDGE set to FALSE.

- To enable the 6 zones in sequence (zone 1 comes on for 10 minutes, zone 2 comes on for 10 minutes after zone 1 is switched off, and so on) the SEQUENCER must be set as follows:

Application examples
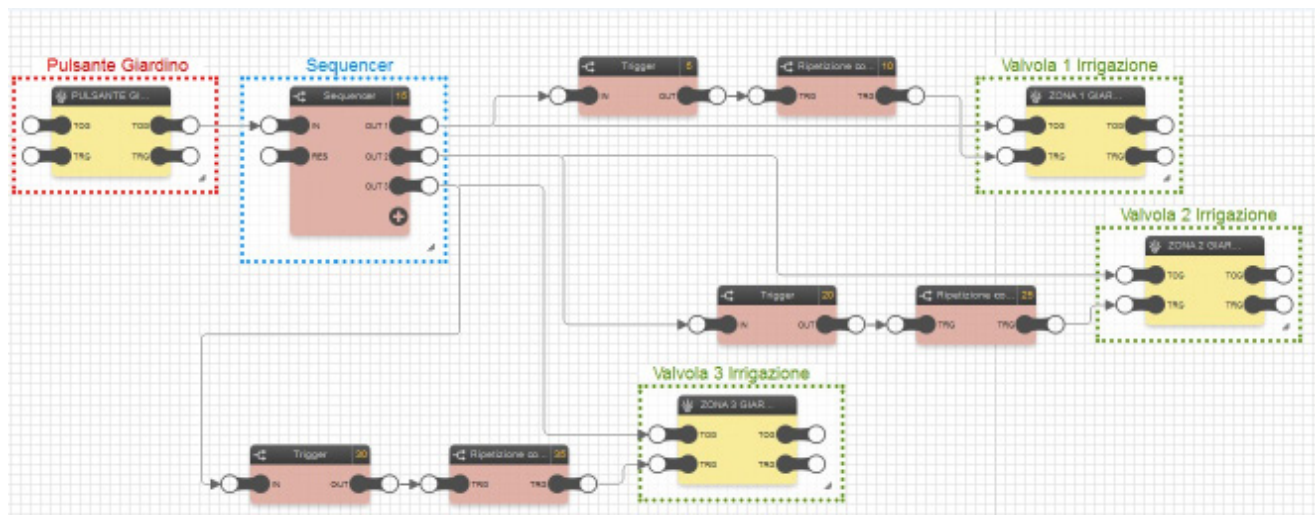
Logic program editor

Application examples

9.3 Sequential and timed irrigation with OFF 3 times for each zone

The logic program switches on the irrigation of a garden, switching on the three zones into which it is divided in sequence.
With the RIPETIZIONE COMANDO block, the OFF command is sent to the irrigation valve 3 times; this override is accomplished with the certainty that this command is received (thus avoiding losing the message).

- The RIPETIZIONE COMANDO block, preceded by the TRIGGER block to provide a pulse at its TRG input, enables the OFF command to be repeated 3 times at 1 s intervals.
- To enable the 3 zones in sequence (zone 1 comes on for 5 seconds, zone 2 comes on for 5 seconds after zone 1 is switched off, and so on), the SEQUENCER must be set as follows:

Logic program editor

Application examples

• The RIPETIZIONE COMANDO block is set as follows:

Proprietà generali	
Tipo:	Ripetizione comando
Ordinamento:	Automatico
Id:	6305
Intervallo(s):	1
Ripetizioni:	3
Elimina	

Proprietà generali	
Tipo:	Trigger
Nome:	TRG
Valori possibili:	0:Off, 1:On
Tipologia nodo:	Trigger
Nodi coinvolti:	LONG.P DIR VAL TOG
Id:	767
Elimina	

Logic program editor

Application examples

9.4 By-me rocker switch used for 2 separate ON/OFF functions

Using a By-me rocker switch, the logic program can manage 2 separate on/off switches on the same button.
The upper button is used for ON-OFF of a utility and the lower button for the ON-OFF of the other.
This solution expands on the functions of the By-me controls which, with conventional configuration, cannot implement the application illustrated here.
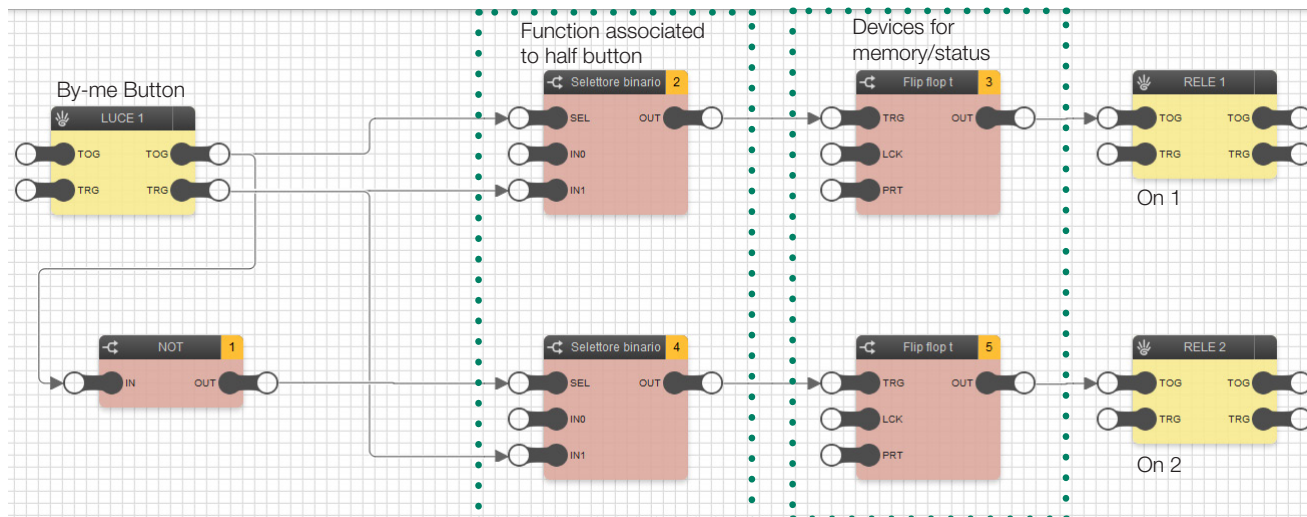

- Each time the button identified by the LUCE 1 block is pressed, the control issues a TRIGGER pulse on the BUS and a control which, according to whether the top or bottom of the button is pressed, may be respectively for group ON or OFF of the group it was configured in.
- The Flip flop T toggles the output state whenever it receives an input pulse, and therefore leaves enabled, or disables when a pulse is received; in this case there must however be a control to ensure that the pulse ONLY arrives if generated by pressing the correct part of the button (top or bottom).
- For this purpose, Selettore binario blocks are used to allow the pulse to pass ONLY if the group control generated with the pulse is correct.

  Example: By pressing the top of the button, Selettore binario 2 allows the pulse to transit since with TOG at 1, SEL also goes to 1, thus accepting the TRG pulse that was set off at the same time and connected to IN1 on the Selettore binario; when it reaches the Flip flop T it will then switch to the previous status.

9.5 Opening/closing roller shutters to preset positions

Via a scenario, the logic program raises the shutters to an intermediate position (different from shutters full up or full down).

Logic program editor

Application examples
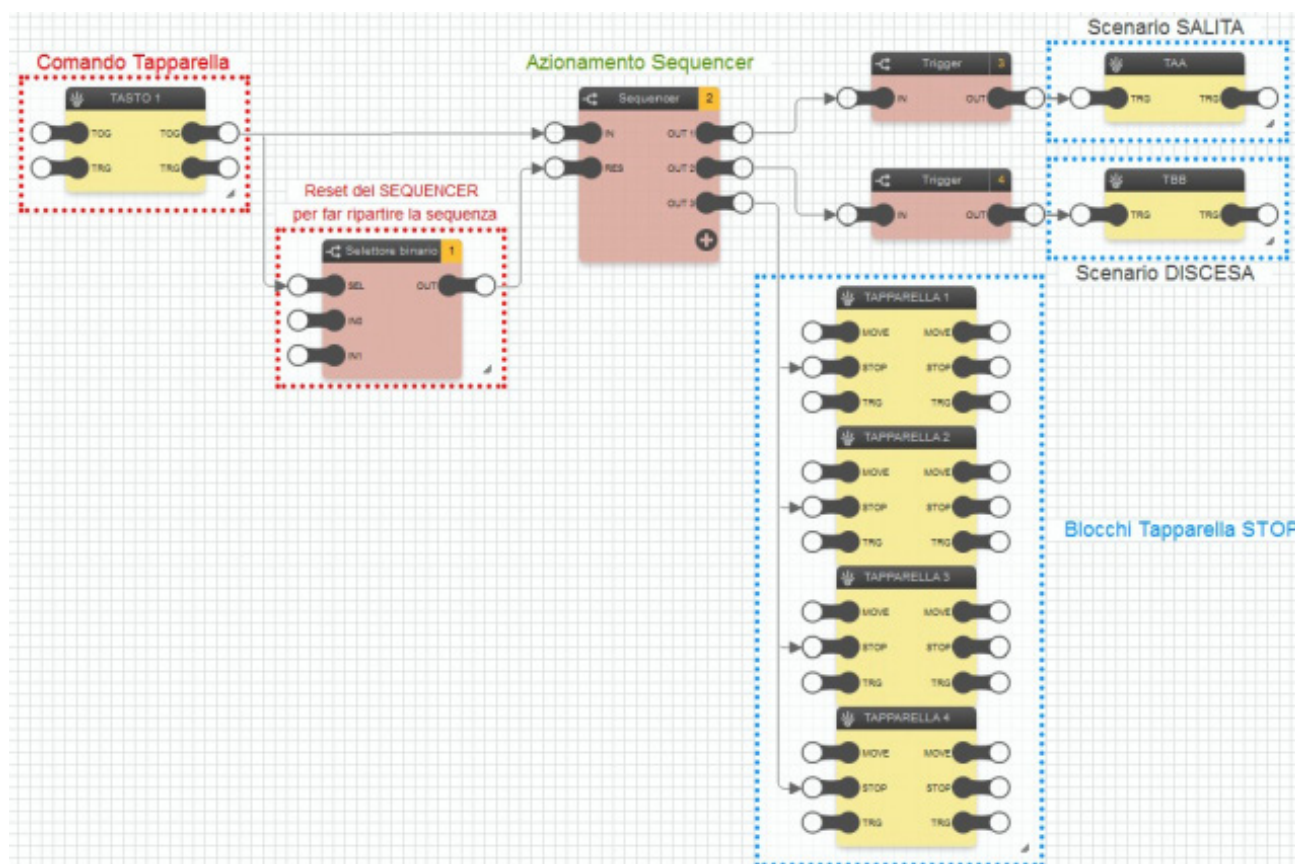
CAUTION: This program functions correctly only if all the shutters concerned have the same up/down times.

• When the input group is ON (TASTO 1 block) the SEQUENCER IS ENABLED; to operate, it enables the output and keeps it active for the time set in the general properties of the block, and then passes to the next until the last output is disabled.

• As outputs OUT 1 and 2 on the SEQUENCER must in any case control the scenarios, the command sent by them cannot be a stable output but rather a pulse; by inserting a TRIGGER block between the outputs OUT 1 and OUT 2 and the respective scenario blocks, this condition is assured. (The Fronte parameter in the TRIGGER is set to TRUE because it must intercept the output activation).

IMPORTANT: When setting the general properties of the SEQUENCER, the time value entered on the OUT1 and OUT 2 step duration is very important, as this determines:
- Durata passo 1: the time taken by the shutters to be raised COMPLETELY must be entered.
- Durata passo 2: the time taken by the shutters to reach the REQUIRED POSITION must be entered.

• The STOP command and the input of the TAPPARELLA blocks is a Boolean value.

Application examples

Proprietà generali	⬍
Tipo:	Sequencer
Ordinamento:	Manuale
Ordine:	
Id:	53688
Sequenza ciclica:	Falso
Durata passo 1:	00:02:00
Durata passo 2:	00:01:15
Durata passo 3:	00:00:01
Durata passo 4:	hh:mm:ss
Durata passo 5:	hh:mm:ss
Durata passo 6:	hh:mm:ss
Durata passo 7:	hh:mm:ss
Durata passo 8:	hh:mm:ss
Durata passo 9:	hh:mm:ss
Durata passo 10:	hh:mm:ss
Elimina	

Ingressi	⬍
IN	Inizio sequenza
RES	Reset sequenza

Uscite	⬍
OUT 1	Uscita 1
OUT 2	Uscita 2
OUT 3	Uscita 3
+	Aggiungi uscita

- The output from OUT3 of the SEQUENCER controls the STOP of all shutters involved in the scenario (in this case 3 groups); between OUT3 and the STOP inputs of the tapparella blocks is the SELETTORE NUMERICO as the STOP node on the tapparella block requires a numerical 1, not a binary 1.
  If "1" is entered as the IN1 value of the Selettore binario, it produces that output value when the command reaches SEL.
  Durata passo 3 (OUT3) can be set to 1 sec as a pulse is sufficient to stop the shutters.

- The SELETTORE BINARIO connected to the RES input of the SEQUENCER ensures that when the input group goes to OFF, the SEQUENCER is reset and restarts (for example, if the user wishes to cancel the program after having commanded it to start). Without the Selettore binario, the Sequencer would stop the sequence and would RESUME FROM THE SAME POINT at the next ON signal (which could cause malfunctions).

- On the SELETTORE BINARIO IN0=1 and IN1=0.

Logic program editor

Application examples

9.6 Switching on outdoor lights from the dusk/dawn sensor and button control

The logic program switches on the outside lights between 21:00 and 5:00, only when triggered by a dusk/dawn sensor.
We can also switch on the lights manually from the button.
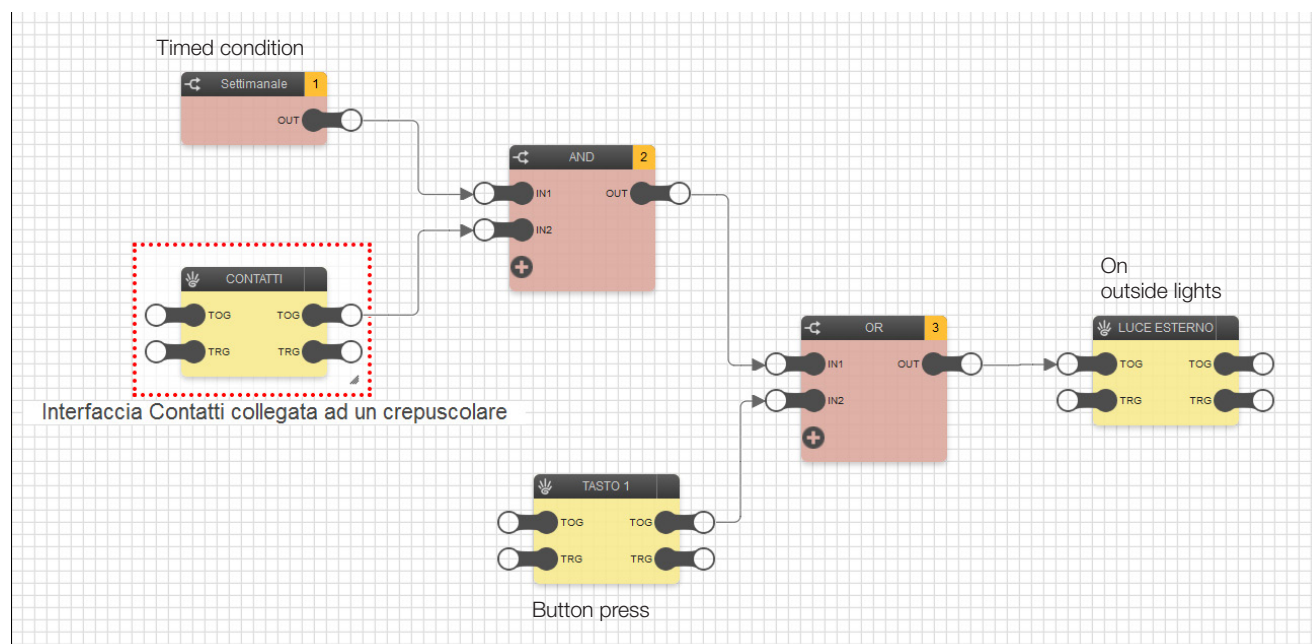
- We use the CONTATTI block (which represents the interface to which the dusk/dawn sensor is connected) and the CRONO SETTIMANALE block, connected respectively to inputs IN1 and IN2 of the AND logic; a scheduling is set in the CRONO SETTIMANALE in which the block is ON every day from 21:00 to 5:00.
- The TASTO 1 block represents the button used to switch the outside lights on whatever the status of the AND logic output (and therefore the reading of the dusk/dawn sensor or the time setting).
- The OR logic is connected to the LUCE ESTERNO block input and ensures that the lights can be switched on from the TASTO 1 block whatever the status of the AND logic.

Logic program editor

Application examples

9.7 Switching on single lights at set times

The logic program switches on the outside lights every night at 1 and at 3 o'clock.
This application is useful, for example, to switch off lights which could remain on all night by mistake.

Evento 1 -- 01:00 ON
Evento 2 -- 03:00 ON
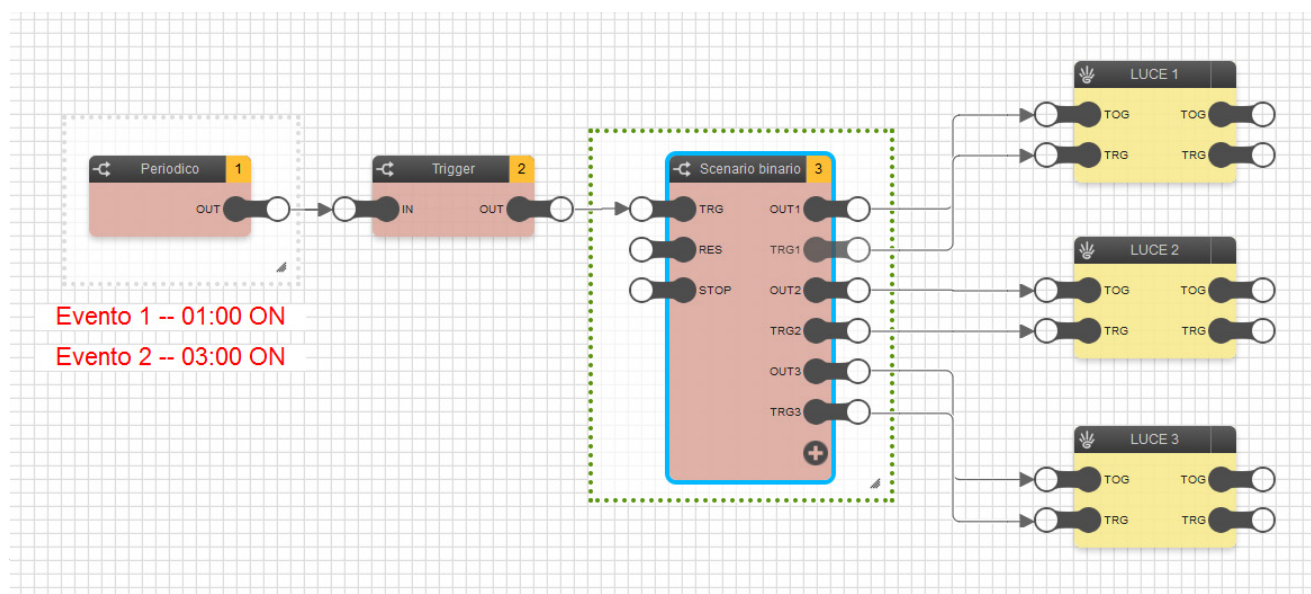
- To create the event in the required time interval, use the PERIODIC TIMER block, to set a scheduling in which Event 1 starts at 1:00 (and Ends, for example, at 2.00) and Event 2 starts at 3.00 (and Ends, for example, at 4.00).
- The CRONO PERIODICO block is connected to a TRIGGER block; this is used to obtain a pulse at the SCENARIO BINARIO block input, the outputs of which OUT and TRG are used to control the respective LUCE blocks.
- The SCENARIO BINARIO blocks must be set as follows to control the LUCE blocks:

9.8 Consumption management for activating the heat pump

When the photovoltaic system produces and feeds into the network 1 kW more than the total consumption, the logic program switches on the load (heat pump) controlling the climate in a room.
Disable as follows:
- if it falls in the band F23 of the contract, the load is kept active for as long as the consumption remains below 500 W;
- the load is switched off if this value is exceeded.
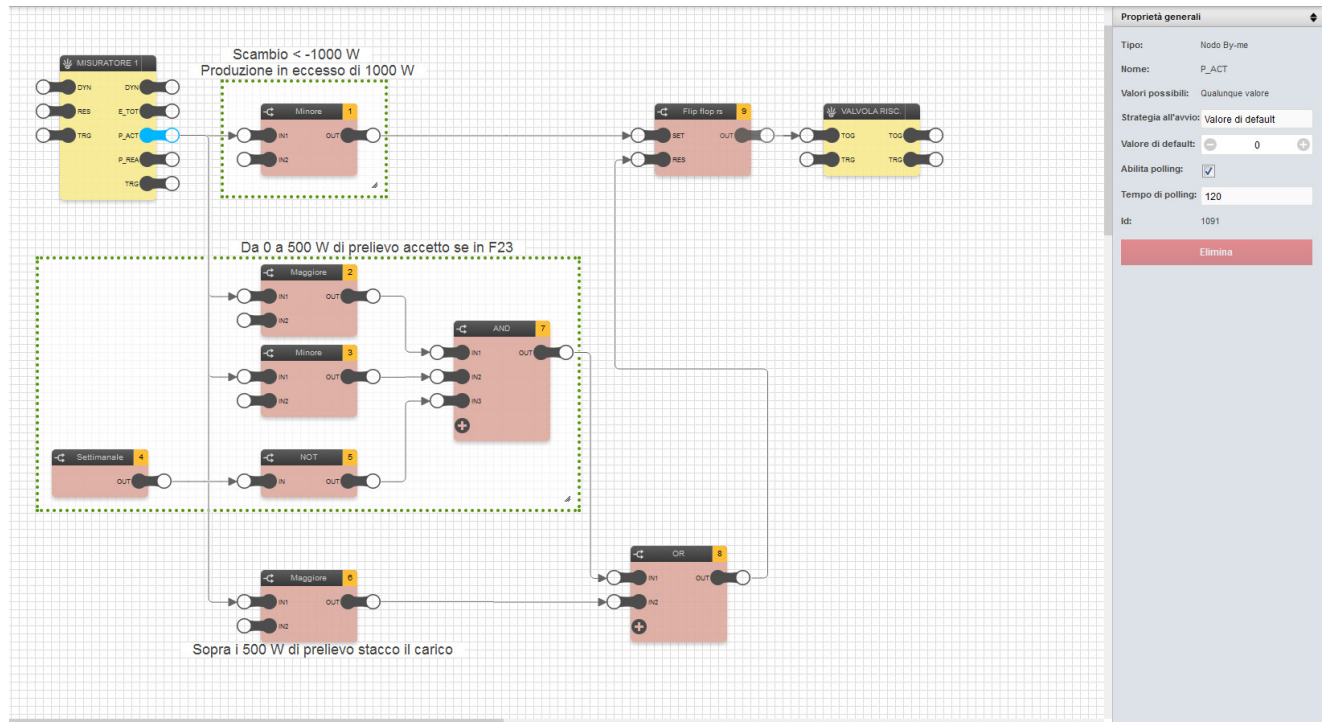
- When a value lower than -1000 W is measured on the node P_ACT, which is the node which views the power exchanged on the electricity supplier meter, there is in fact an excess production of over 1000 W; if this condition occurs, the MINORE 1 block, in which the other node is set to -1000, sets the OUT output to TRUE. The latter, controlling the SET input of the FLIP FLOP, sets OUT to 1 and therefore controls the VALVOLARISC load.

- The FLIP FLOP remains on until TRUE is given on the RES input; then all the other logics which define the behaviour if there is no excess production come into play (i.e. P_ACT>=0).

- The logic blocks contained in the larger box mean that, if the consumption is between 0 and 500 W in the low-cost time band F23, the load is in any case kept on.
  In fact, the MAGGIORE 2 block becomes TRUE as soon as P_ACT exceeds 0, MINORE 3 becomes TRUE if P_ACT stays below 500 W while the SETTI-MANALE connected to the NOT block sets a TRUE value only OUTSIDE the F23 time band, otherwise it would be necessary to switch off the load (we would be in band F1, which has a higher cost); thus, if all three conditions are TRUE, the AND block sends TRUE to the OR block, which in turn sends it to the RES node on the FLIP FLOP, which resets the output and switches off the load.
 The comparison blocks can be set as follows:
  - MINORE 1: IN2=-1000
  - MAGGIORE 2: IN2=0.
  - MINORE 3: IN2=500.
  - MAGGIORE 6: IN2=500

- The MAGGIORE 5 block, which becomes true if P_ACT exceeds 500 W and immediately switches off the load, is connected directly to the OR block. If this is TRUE, as above, it sets the RES node to TRUE, which resets the FLIP FLOP output.

Logic program editor

Application examples

9.9 Dehumidifier management via several humidity probes

Using more than one humidity probes in the same system, the logic program controls a single dehumidifier according to the values read by each probe.
As By-me systems are designed to be managed by only one humidity probe, this limit is overcome using the logic program.


Comparison of measured value/ threshold

Comparison of measured value/threshold

Dehumidifier
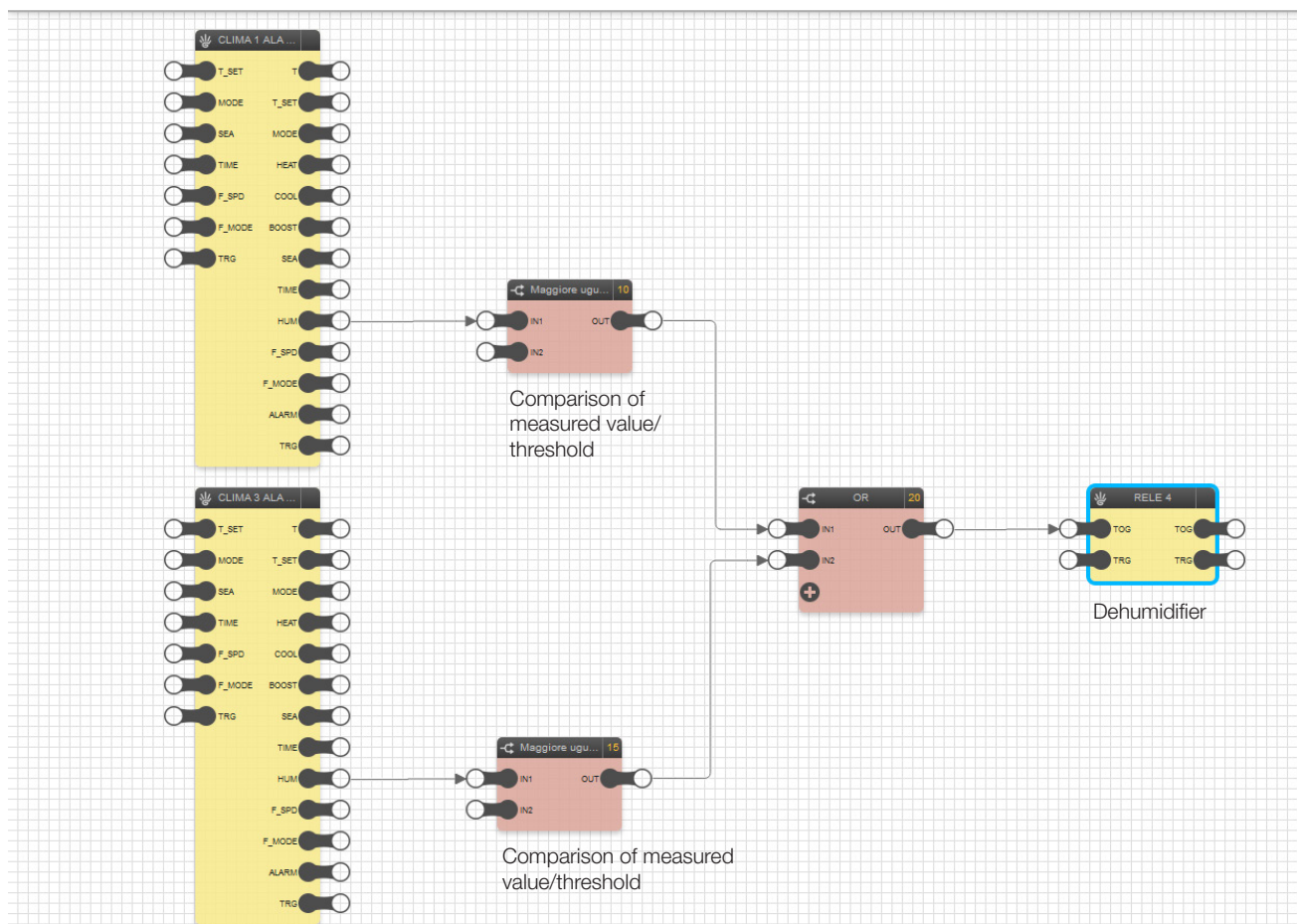
- The CLIMA block is used to represent each of the probes present in the system (HUM output).
  In this example the humidity data is provided by a climate zone in which, as well as the thermostat, an input is configured in the group connected to a humidity sensor.
- Each HUM output is connected to the MAGGIORE UGUALE comparison block; if the value at the input IN1 is greater than or equal to that set on IN2, 1 will be obtained at OUT (ON).
- All the comparison blocks are connected to the OR logic block, since to switch on the RELE block representing the dehumidifier, it is sufficient for just one of the probes to measure a humidity value greater than or equal to that set.

Application examples

9.10 Multiple activations from a single control.

Via a single control, the logic program performs multiple activations, which not only control the group ON/OFF modes but also control the groups (for example, audio output with volume at 90% and lights at 50%).
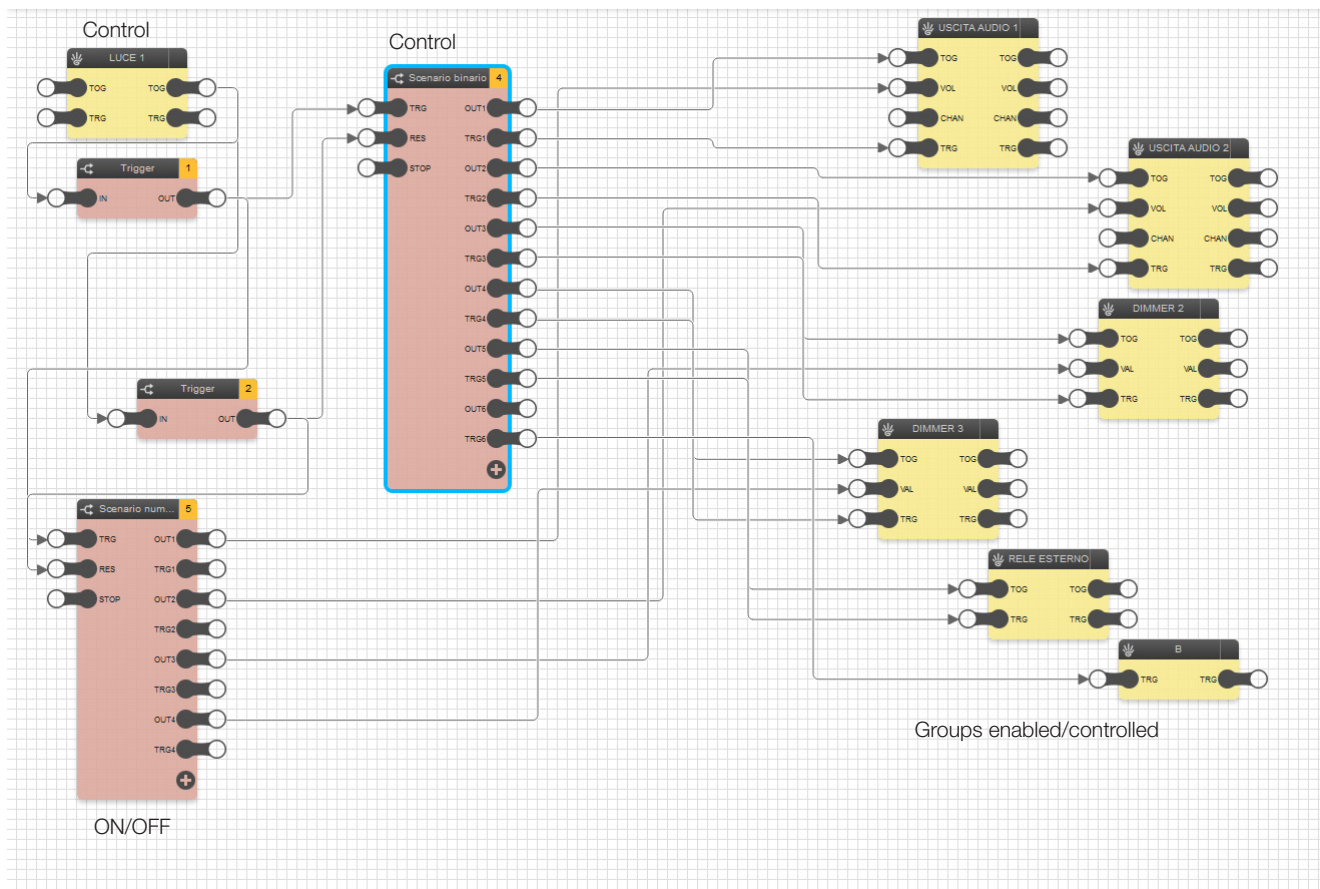
- When a command is sent from the LUCE 1 block, the TRIGGER 1 block transits a pulse only if it has received 1 at the input, while TRIGGER 2 sends the pulse only if it has received a 0.
- TRIGGER 1 transfers the pulse to the TRG input of the SCENARIO BINARIO and SCENARIO NUMERICO blocks, activating the various outputs of both scenarios; TRIGGER 2 on the other hand sends a pulse to the RES input of the two SCENARIO blocks, restoring all output values from them to 0.
- Some blocks have the option of being controlled by both scenarios as there is a need to activate them and to control their operation (for example, volume in the audio zone, brightness of the dimmer); therefore, commands sent from the SCENARIO NUMERICO and the SCENARIO BINARIO will reach the inputs of the same block at the same time.
- The SCENARIO BINARIO block is set as follows to command the blocks USCITA AUDIO 1 and 2, DIMMER, etc.:

Application examples

Logic Unit

Application examples

9.11 Disabling of loads with delayed enabling (planned manually on the load in time bands with probable PHV production) when the available energy is not sufficient to power them.

Modern household appliances can be programmed to come on automatically at a given time; these programs are however cancelled if the power to the appliance is switched off for a certain period of time. If at the time the appliance is switched on there is not at least 1000 W of excess power, the logic program illustrated forces the relay connected to the appliance to OFF so that the set program does not run, thus avoiding consumption of energy from the mains electricity.


- A MISURATORE block is used in which, setting a polling time of 120 s on the output P_ACT, it updates the power data measured on the MAGGIORE block input every 2 minutes.

- The MAGGIORE block is set so that its OUT is TRUE only if the value measured by P_ACT is greater than -1000 W (for example -900) and indicates that the power produced but not used on the system does not exceed 1000 W.

- The AND block output becomes ON only if this condition is valid in the time interval set on the PERIODICO 3 block.

- The scheduling on the PERIODICO 3 block is set to obtain activation only for 5 minutes prior to the appliance being switched on; in this way, the control is run only in this time interval, while outside of this it does not interfere with the load function.

- When the AND block goes to ON, the OUT on the FLIP FLOP RS goes to ON and remains in that state even after the block AND returns to OFF; this means that the load remains forced OFF (the OFF input of the load is kept on by means of the SELETTORE NUMERICO) until the FLIP FLOP RS block is reset.

- The FLIP FLOP RS is reset by sending an ON to the RES node via the block PERIODICO 5; the scheduling sends ON to a time after that of the block PERIODICO 3 and in a sufficiently large time interval to ensure that the load has been switched off and the programs set on the appliance deleted.

Logic Unit

Application examples

9.12 Forced OFF command repeated 5 times

The logic program enables RGB dimmer management with a By-me command and a traditional dimmer.

With the RIPETIZIONE COMANDO block, the OFF command is sent to the RGB dimmer 5 times; this override is accomplished with the certainty that this command is received (thus avoiding losing the message).
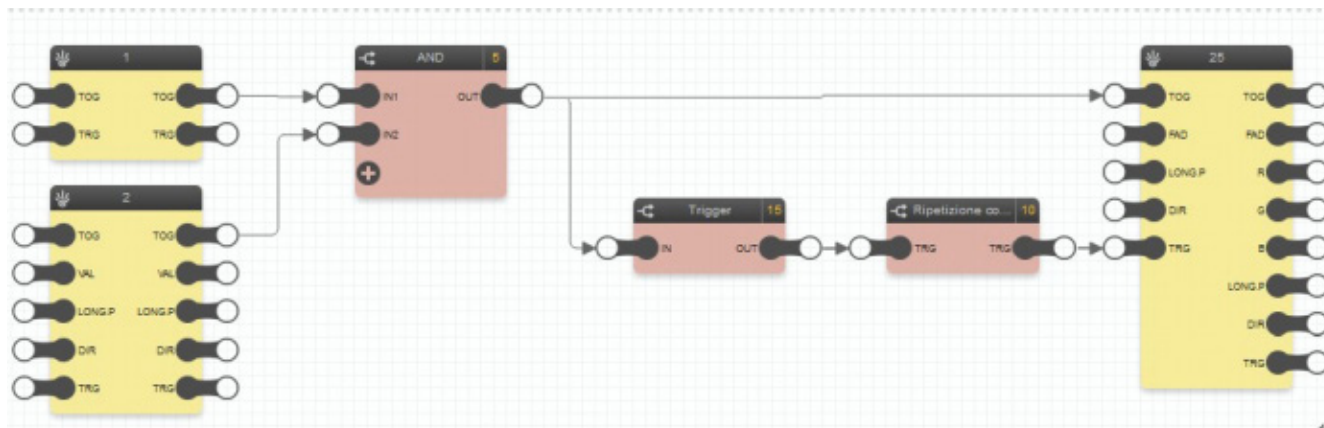
• You use the block 1 (COMANDO BY-ME) and block 2 (DIMMER) to control the RGB block by connecting their inputs to the AND block.

• The RIPETIZIONE COMANDO block, preceded by the TRIGGER block to provide a pulse at its TRG input, enables the OFF command to be repeated 5 times at 1 s intervals.

• The RIPETIZIONE COMANDO block is set as follows:

CE

CE